COMPUTER ANIMATION OF CONJUGATE SURFACES

by

P. S. AVADHANI

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, KANPU
SEPTEMBER 1993

COMPUTER ANIMATION OF CONJUGATE SURFACES

A Thesis Submitted
In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF TECHNOLOGY

by
P.S.Avadhani

to the
Department of Computer Science and Engineering
INDIAN INSTITUTE OF TECHNOLOGY KANPUR
September, 1993

CERTIFICATE

It is certified that the work contained in the thesis entitled COMPUTER ANIMATION OF CONJUGATE SURFACES by P.S.Avadhani, has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Dr.Sanjay G.Dhande,
Professor,
Department of Computer Science and Engineering,
I.I.T. KANPUR.

September, 1993

Dedicated To

Duru

ACKNOWLEDGEMENTS

I would like to express my deep gratitude to my guide, Dr.Sanjay G.Dhande for his expert guidance and constant encouragement throughout this thesis work.

I am grateful to Mr.K.P.Karunakaran, who, despite his busy schedule, always helped me whenever I was in trouble, both in academic and non academic matters.

I would like to place on record my sincere gratitude for the authorities of Andhra University, Visakhapatnam for sponsoring me to this program without which this work would not have been possible.

I am grateful to Prof. S.Biswas, Coordinator, CMP for his help.

I would like to thank the lab staff of CSE department and the software engineers at the computer center for their cooperation during this work. Special thanks to Mr.N.V.Brahmaji Rao, Software Engineer, CSE department, for his help in the preparation of this document.

I thank my friend Mr.M.V.R.Gopal for his helping me in debugging my programs.

I am thankful to my friends, Mr.P.V.M.Rao, J.venkatesh, Gagan Saxena and B.S.S.Pradhan for the fruitful discussions I had with them.

I am grateful to Mrs.S.Biswas of CAD-P and to my friends Mr.P.V.S.N.Murthy, Mr.S.P.Chandrasekhar, Dr.V.C.V.Rao and Mr.Appaji Rao for their company. I am thankful the staff of the CAD-P for their help .

My special thanks to Mr.P.V.Ravindra and Mrs.Sunitha Ravindra for the delicious Sunday dinners I had in their house during the last two months of my stay .

I could not have completed this work without the support of my wife, Lakshmi, who, despite her own domestic troubles kept on encouraging me.

— **P.S.Avadhani.**

Contents

List of Figures

ABSTRACT

Computer Animation has been a topić of research for many scientists involved with human body modelling and movement, education and visualization of mathematical and scientific models. The main thrust of most of the research has been towards the human body modelling.

The report proposes a methodology for animating two moving surfaces in higher pair contact (called the conjugate surfaces). The methodology is based on a symbolic algorithm which has been implemented using a symbolic computation program MACSYMA. Two case studies, namely, a 2D cam and a 3D cam with translating followers have been shown in animation using this algorithm to illustrate the model.

A general model of the cutter is proposed and the surface generated by the cutter is shown in animation from a specified initial point to a specified final point. Linear interpolation is used as the in-betweening technique.

The present work can be applied in Design and Manufacturing field as it illustrates the design of the cutter surfaces.

This work is implemented using the starbase graphics library on HP-9000/834 Turbo SRX 3D graphics work stations.

Chapter 1

INTRODUCTION

Recent developments in Computer Graphics have opened new dimensions in the fields of Animation, Graphics, Design and Scientific Visualization. Primarily for animation, handling of huge data sets and the speed are the basic problems. This is to some extent being solved at the advent of the high speed computers with high data handling capabilities. It is also to be noted here that the geometry of the object or scene to be displayed in animation plays another important role. The motion dynamics[16], the in-betweening techniques[22] add to the realism in animation[43]. For example, surface generation is a major area of interest in solid modeling, design and manufacturing science. Suppose, a cutter generates a surface while cutting a blank, then to show that surface with cutter in animation, we should know the geometry of the cutter, the geometry of the surface to be generated and the mode of in-betweening apart from the relative motion specifications. In fact, the developments in geometric and solid modeling techniques have provided the required base for these problems for design and manufacturing engineers with an elegant and precise definition of a component or product that needs to be designed[31,64,67]. In manufacturing science several methodologies based on these modeling techniques were developed. Recently a unified methodology for geometric modeling of various surfaces using the concept of Conjugate Geometry is developed. This methodology is useful in developing any two surfaces in higher pair contact and in movement. The work proposed in this thesis is to display the cutter from a specified initial position to a specified final position generating the surface in motion using the symbolic algorithm based on Conjugate

Geometry. The following sections illustrate the basic concepts of Conjugate Geometry and animation.

1.1 CONJUGATE SURFACE GEOMETRIES:

It can be seen from the experience that a mathematical description of the shape of an object will be specified depending on the purpose of its use. For example, if a shape is to be displayed on a computer controlled display device then a parametric definition is appropriate whereas if a shape is to be specified for some analytical procedure then a closed form expression in Cartesian coordinates is used. However, if a shape is to be realized by generating process then neither of these is useful.

Consider the case of generating a circle from a 2D lamina. One way could be to employ a series of straight line cuts (see Fig.1-1). These cuts constitute a family of curves. An envelope to this family is the shape generated , which in this case, is a circle. Similarly, a straight line can be realized as an envelope of a family of circles. The family is generated by moving the center of the circle on a linear path. Thus, a family of circles generates a linear shape and a family of lines generates circular shape (See Fig.1-2). Thus envelope theory is the mathematical basis for modeling the geometry of several shape generating processes. Traditionally the envelope theory defines an enveloping curve or surface in a fixed frame of reference. However, many manufacturing processes require moving frames of references. The conjugate geometry model [28] extends the classical envelope theory and defines the envelope in one moving frame of reference due to a family generated by another shape (curve or surface) and the relative motion between the two frames (see Fig.2-1). The present work uses the conjugate geometry approach to simulate the animation of a cutter to generate a surface from a given initial position to a given final position.

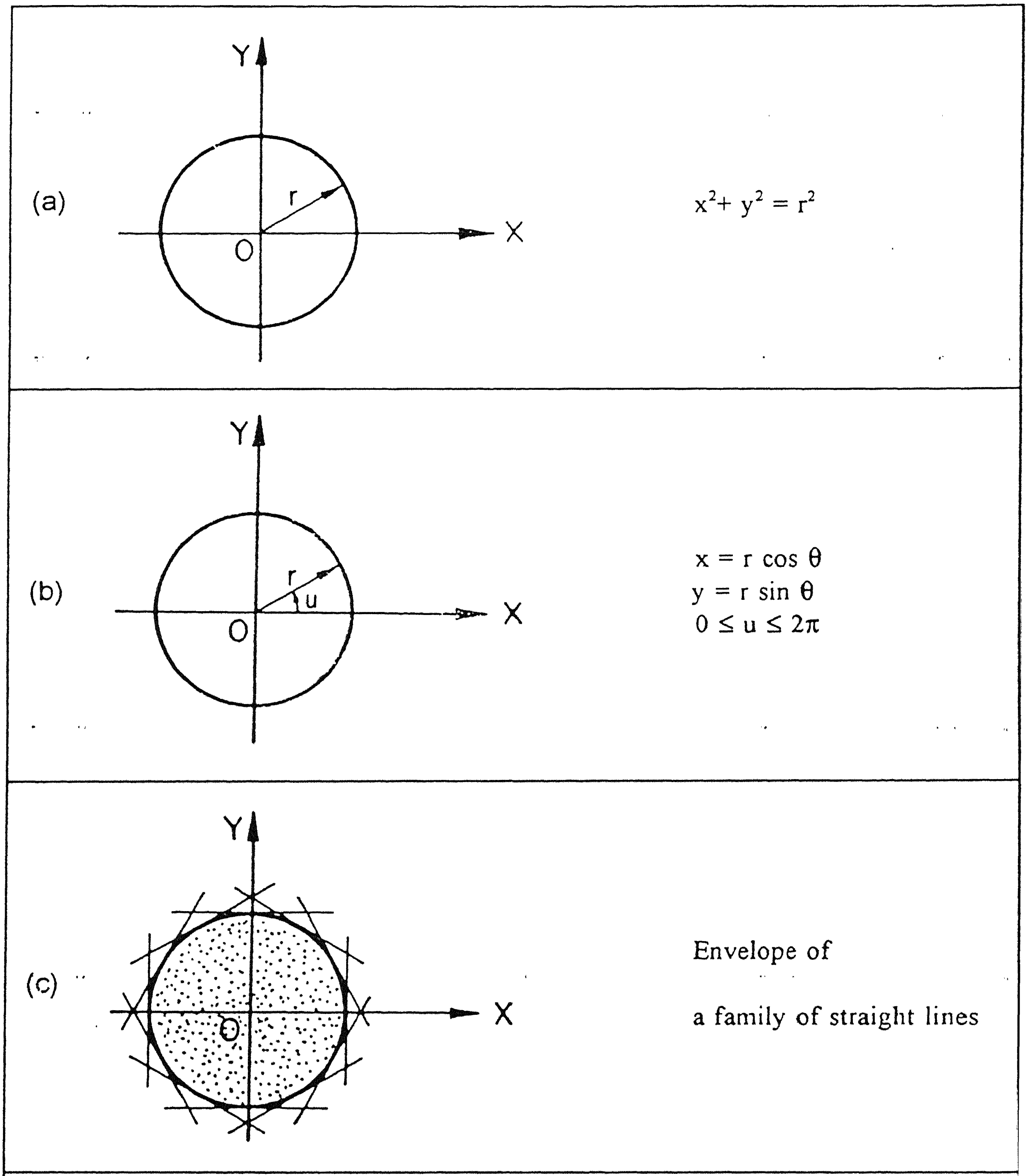

Figure 1-1 Definition, Display and Realization of a Circular Shape
(a) Mathematical Definition - Implicit Form
(b) Display Definition - Parametric Form and
(c) Realization Definition - Envelope Form

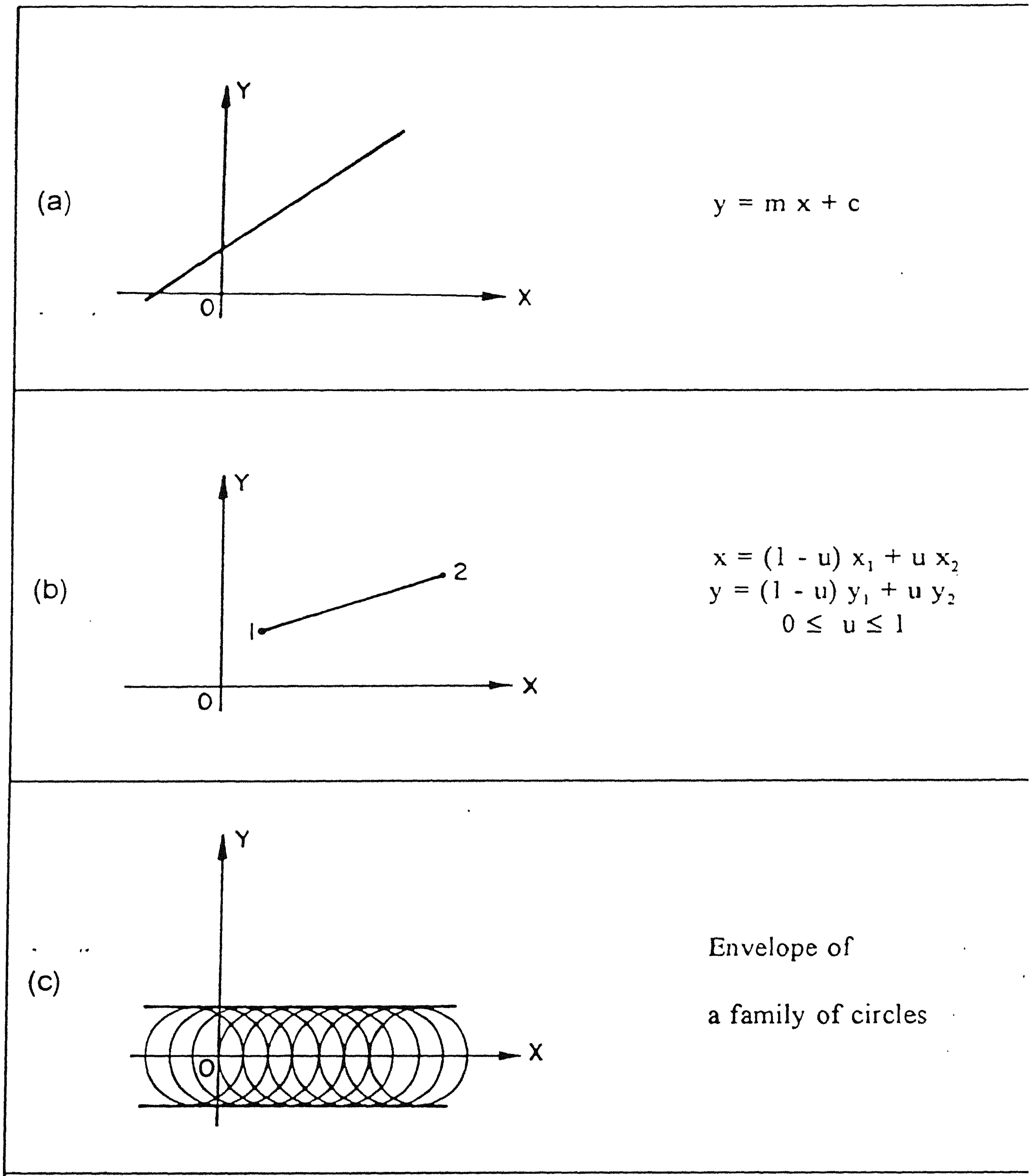

Figure 1-2 Definition, Display and Realization of a Straight Line Shape
(a) Mathematical Definition - Implicit Form
(b) Display Definition - Parametric Form and
(c) Realization Definition - Envelope Form

1.2 COMPUTER ANIMATION:

Animation means bringing to life. In this view, people often think of animation as synonymous with motion. However, it covers all changes that have a visual effect. Thus, it includes the motion dynamics (i.e. the time varying positions) and update dynamics (shape, color, transparency and texture) and all changes of rendering effects of an object. Animation is used widely in the entertainment industry. At the advent of computers, it is also being applied in education, in industrial applications such as control systems, flight simulators for aircraft and in scientific research. The application of Computer Graphics and especially of animation have come to be grouped under scientific visualization. Often, the animations in scientific visualization are generated from simulations of scientific phenomenon. The results of the simulations may be large data sets which are converted into images that constitute the animation. Visualization is more than mere application of graphics to science and engineering and it can involve other disciplines such as Computational Geometry and Geometric modeling. The present work is to show a manufacturing process in animation. All manufacturing processes are shape realization processes. The shape realization is equally important as its design. The proposed animation enables a manufacturing engineer to evaluate the likely outcome of a particular cutter of a manufacturing process.

Conventional animation is oriented towards the production of two dimensional cartoon films and it is a fairly fixed sequence. The steps involved in this are

1. Story:

a) Synopsis - The summary of the story in a few lines.

b) Scenario - Detailed text that describes the complete story without any cinematographic references.

c) Story board - This is a film in outline form. It consists of a number of illustrations arranged in comic-strip fashion with appropriate captions. The number of individual illustrations within a story board will vary widely. The important aspect here is that they represent the key moments of the film. I is also important to note that the film is composed of sequences that define specific actions. Each sequence consists of a series of scenes that are generally defined by a ertain location and set of characters. Scenes are divided into shots that are considered as picture units.

2. Layout - This step consists of the design of the characters to be animated and action plotting. Based on the story board the relationship between the shapes and the forms in the background is decided. The layout department in an animation studio has to finish drawings of the settings and sketch background layouts. The artists must have a knowledge of the physical characteristics of the camera that will be used to photograph the frames.

3. Sound track - In conventional animation, sound track recording has to precede the animation process because the motion must match the dialogue and /or music.

4. Animation - The animation process is carried out by the animators who draw key frames. Often an animator is responsible for one specific character.

5. In-betweening - "In-betweens" are defined as drawings which are placed between two key positions or frames. Assistant animators draw some in-betweens and in-betweeners draw the remaining figures. The work of the assistant animators requires more artistry than the in-betweeners, whose task is almost automatic.

6. Xeroxing and Inking - Sketches are usually drawn and then they have to be transferred to acetate cels using modified xerox cameras. Lines will be inked in by hand.

7. Painting - As cartoon animated films are usually in color, they must go through a painting stage. This work requires patience and accuracy. Cels must have the right degree of opacity and static backgrounds must also be painted.

8. Checking - Animators need to check the action in their scenes before shooting.

9. Shooting - The final photography of composite animation is usually done on color films or video tapes. The shooting phase is not a trivial operation. Movement can be simulated at this stage by moving certain cels with respect to some others. Many cels are placed in layers one above the other and are shot simultaneously.

10. Editing - This step is considered part of the post production stage.

Introduction of computer in the field of animation came as more than automating the various steps in the traditional animation process. The initial application was in the in-betweening, coloring and copying phases. But soon it developed into an entirely new field by itself. This was brought about mainly by the ability of the computers to represent, process and operate on objects of dimensions higher than two. Now the main task lies in modeling, representing and simulating dynamic phenomena of all kinds.

Most of the phenomena which may be represented on the screen of a workstation are typically time-dependent. The techniques of Computer Graphics allow construction of 3D graphical objects using geometric modeling techniques. Moreover, in a 3D space, scenes are viewed using vertical cameras and they may be lit by synthetic light sources. In order to visualize these phenomena at any given time, it is necessary to know the appearance of the scene at this time and then computer graphic techniques allow us to build and display the scene according to viewing and lighting parameters. The general problem in this situation is to express time dependence in the scene and how to make it evolve over time. These problems and their various solutions are part of what is usually understood by the term Computer animation. Thus computer animation may be defined as a technique in which the illustrations of movement is created by display on a screen, a series of individual states of a dynamic scene. Broadly, there are three ways of animating this ,namely, rotoscopy, key frame animation and procedural animation.

In key frame animation, a certain number of frames called key frames are given to the system and the computer derives the other frames using interpolation procedures.

Various approaches have been developed in this process. The two most important and widely used techniques are shape interpolation and parametric key frame animation. In shape interpolation the in-betweens are obtained by interpolating the key frame images themselves. In parametric key frame animation objects are characterized by parameters; motion is specified by giving key values for each parameter; in-between values are calculated using an interpolation law. In procedural animation, motion is algorithmically described.

Initially and even now, most of the in-betweening work is done using the technique of linear interpolation. However, it is observed that this might cause a lack of smoothness which considerably alters the motion. Undesirable effects such as discontinuities in the speed of motion and distortions in rotations are produced. Consequently for such motions, other methods should be used. A non-linear interpolations (non linear in time but not in space) like spline interpolations are some of the methods tried by researchers with good results.

1.3 LITERATURE REVIEW:

The idea of animation is conceived as early as 1831 by Joseph Antoine Plateau who produced a device called phenakisto scope. This consisted of a spinning disk with drawings that could be viewed through framing windows. Almost at the same time, Horner produced a device called the Zoetrope, in which drawings were produced on the inner surface of a drum. The individual frames could be viewed through slits in the inner rotating drum as they came past a fixed framing slit, through which the viewer looked. Devices of this sort became highly popular during the last half of the previous century.

Paralleling this development in frame animation, photography evolved fast in the last half of nineteenth century giving boost to the field of animation. In 1915, Earl Hurd introduced the technique of cel animation which took its name from the transparent sheets of celluloid that is used. But, of course, the father of commercial animation is certainly Walt Disney . In the ten years 1928-1938 he produced Mickey Mouse Donald Duck and the Silly Symphony series.

The animation film industry was well established when the computer made its first contributions. One of the early pioneers were Bell Laboratories. This work was primarily directed at the scientific and education world. However, it was not long before the importance of the computers to the animation industry was realized. Advances in real-time systems allowed the computer to automate the control of the rostrum camera. Kallis[38] in 1971 described a system capable of carrying out sequences of operations using pan, tilt, spin and zoom. Following this lot of work has been done in the automation of in-betweening.

Reeves [79] proposed a method called moving point constraint method. This method is meant to allow the specification of multiple paths and speeds of interpolation and to reduce motion discontinuities at key frames. The principle of the technique is to associate a curve varying in space and time with points of the animated objects called the moving point and it controls the trajectory and the dynamics of the point. Kochanek-Bartels[51] used spline interpolation to control the tension, continuity and bias and thereby controlling the in-betweens. These methods coupled with methods based on aerodynamics are used to simulate and control the motion in objects in fluid flows[95]. Freedman et al[35] used a technique of quadrature pair of oriented filters to vary the local phase, giving the sensation of movement for displaying patterns that appear to move continuously without changing their positions.

An interactive technique for animating deformable objects is developed in [26]. This approach provides a representation of the deformations independent of the surface geometry and can be easily integrated into traditional hierarchical animation systems.

A general interactive modeling and animation system that facilitates a variety of simulation and animation paradigms like cameras, light sources, rendering and user interfaces have been proposed in [101]. Much work has been done in human body modeling and the pioneers in this area are Thalmann [59,60,61,62,63].

Human hand motion and grasping of arbitrary shaped objects is developed in [82] using a knowledge based approach. This effectively reduces the enormous search space associated with the problem of grasping objects and satisfying the kinematic and physical constraints associated with the problem. A methodology for controlling the dynamic legged motion is proposed in [77]. Lasseter[54] ,Laybourne[56] and Shoup[88] give a very good account of the techniques of traditional animation applied to 3D Computer animation apart from the various other techniques of in-betweening and rendering. Burtnyk[16] proposed techniques to enhance the motion in key frame animation.

Animation of facial expressions was dealt by Parke[74], Platt and Badler[75], Thalmann[58] and Max[65]. Shoemake [87] proposed a method based on quaternions for rotations in animation instead of conventional methods.

The developments in geometric and solid modeling techniques [67, 64] have given the manufacturing engineer the required methodology. The problem of conjugate geometry between a pair of moving as well as contacting surfaces is termed as a higher pair contact problem in kinematics[7, 23, 97]. Such problems have been analyzed by designers of cams and gears. Chakraborty and Dhande[23] described the methodology of defining the kinematics of relative motion as well as the geometry of the contacting surfaces for three dimensional cams. One of the major computational difficulties in formulating and analyzing the problems of conjugate geometry is that for every given situation of a pair of contacting surfaces and their relative motion, one has to derive the biparametric equations of the surfaces every time. This can be improved by using the symbolic manipulation approach [28, 45]. A symbolic model with twelve degrees of freedom is proposed in [28].

1.4 SCOPE OF THE PRESENT WORK:

The idea of the present work is to depict the power of conjugate geometry approach to display conjugate surfaces in animation. Two examples of 2D and 3D cams with translatory motion are displayed in animation using the symbolic algorithm. The next phase can be viewed as two parts: The surface generation and the animation. A generic cutter with nine parameters is defined and displayed. The cutter draws the surface it generates from a given initial position to a given final position while moving along the path. The initial and final positions will have three rotations and three translations as their input. It means that the cutter can translate and/or rotate along any axis while it is moving.

Initially, the cutter is drawn with the given nine parameters(which determine the cutter) using an algorithm. Next, the symbolic program MACSYMA is used to generate the C-code for the conjugate surface generated by the cutter. In fact, it is a set of eight surfaces that are drawn, as the cutter shape has eight segments, namely, six straight line segments and two circular segments. Next, a trimming algorithm is used for depicting the ends of the surfaces generated. Thus the total number of surfaces generated will be ten in number. These ten surfaces are properly combined along with the cutter itself and are generated part by part for each time frame to give the effect of continuous motion.

All these algorithms are implemented using the double buffering technique of starbase graphics library on HP-9000/834 Turbo SRX 3D graphics work stations based on HPPA RISC CPU rated at 2.02 MFLOPS(14 MIPS) performance.

1.5 ORGANIZATION OF THE THESIS:

In chapter 2, a detailed account of the computational conjugate geometry along with the symbolic algorithm are presented. In section 2.4 two case studies, namely, the 2D and 3D cams are presented to illustrate the power of the symbolic algorithm. The animation aspect is also taken into account. Chapter 3 deals with the generic cutter definition, the algorithm to generate the cutter and the methodology adopted to generate the surfaces and animation. Chapter 4 deals with the aspects of software developments and methodology adopted and finally chapter 5 offers conclusions.

Chapter 2

COMPUTATIONAL CONJUGATE GEOMETRY

2.1 SYMBOLIC MODEL:

There are basically three elements involved in any machining process. The first one is the surface of the cutter denoted by Σ_1 ,the second being the surface that is to be produced denoted Σ_2 (see Fig.2-1). The third element is the relative motion between these two surfaces which will be constrained by the type of kinematic chain of the machine tool. The surfaces Σ_1 and Σ_2 are conjugate to each other as those two surfaces maintain a higher pair contact throughout the cutting motion . A unified symbolic model based on the computational conjugate geometry is proposed in [28] with twelve degrees of freedom. Using this symbolic model, if any two of the three elements, namely, the surfaces Σ_1, Σ_2 and the relative motion parameters are known, the third one can be determined. This model is useful in finding the NC cutter path when the surfaces Σ_1 and Σ_2 along with the relative motion parameters are required to be determined. This model is useful in NC path simulation, NC code optimization etc. The best use of this model may be for choosing the optimal combination of the machine tool and the cutting tool by an iterative process.

2.2 BASICS AND OVER VIEW OF THE SYMBOLIC ALGORITHM:

A geometric model of a machine tool is essentially a representation of the kinematic structure of the machine tool . A kinematic structure of a machine tool describes the kinematic chain necessary for providing the cutter motion to the cutter and the feed motions to the cutter and /or the blank. The symbolic model considers the cutter and the blank to be two free bodies in space and thus their position and orientation can be described by six degrees of freedom each. These twelve degrees of freedom are expressed with respect to a fixed frame of reference $S_0(O_0 - X_0, Y_0, Z_0)$. The coordinate systems $S_1(O_1 - X_1, Y_1, Z_1)$ and $S_2(O_2 - X_2, Y_2, Z_2)$ are considered attached to the cutter and blank respectively (see Fig.2-2). The initial positions of S_1 and S_2 are denoted by $S_{\bar{1}}(O_{\bar{1}} - X_{\bar{1}}, Y_{\bar{1}}, Z_{\bar{1}})$ and $S_{\bar{2}}(O_{\bar{2}} - X_{\bar{2}}, Y_{\bar{2}}, Z_{\bar{2}})$.

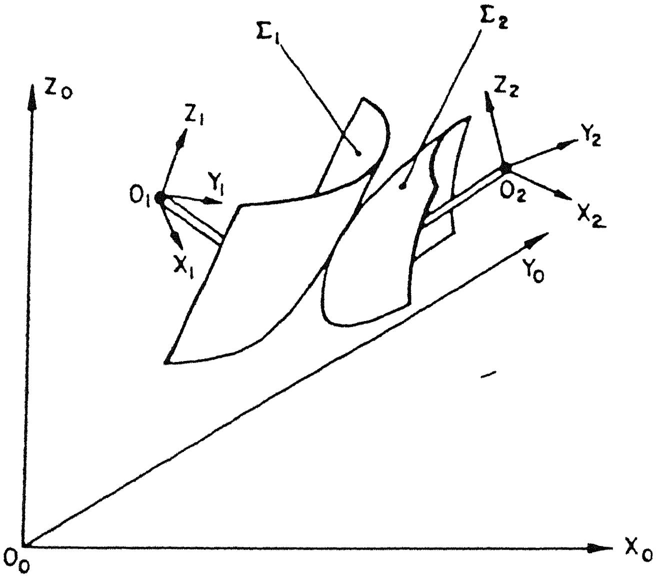

Figure 2-1 A Pair of Conjugate Surfaces

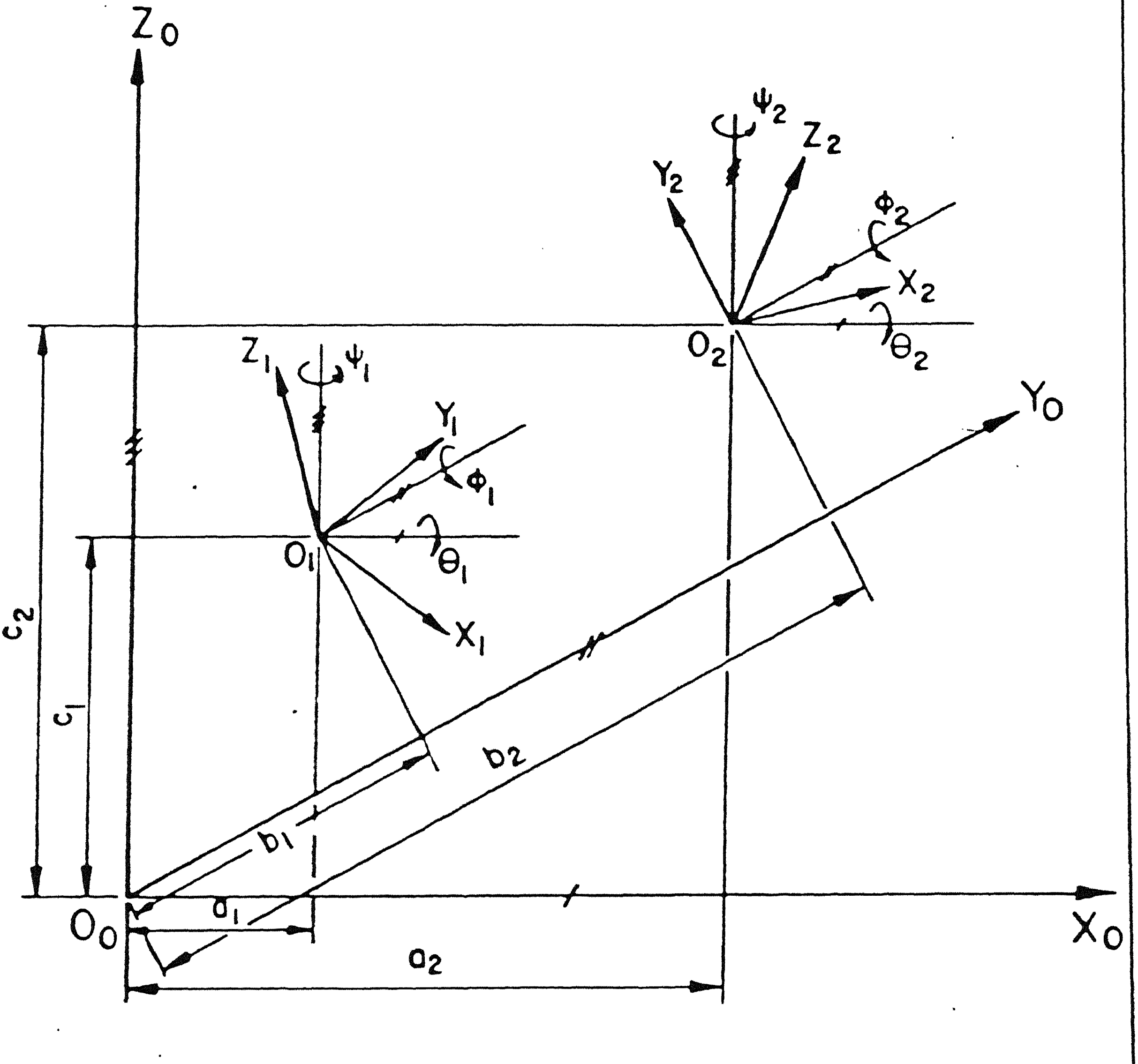

Figure 2-2 Translational and Rotatioinal Degrees of Freedom of Conjugate Coordinate Systems

Let the six tuple $(a_i, b_i, c_i, \theta_i, \phi_i, \psi_i), i = 1, 2$ denote the translations and rotations of the moving coordinate frames S_1 and S_2 respectively. These twelve parameters of motion represent the kinematic arrangement of the machine tool so that they can be expressed as functions of time t. As the tool surface Σ_1 and the surface Σ_2 produced by blank are time-invariants, the known surface among the two surfaces can be expressed as a biparametric surface in u and v. Since the two Σ_1 and Σ_2 surfaces are in higher pair contact, at any point of contact P,

$${}^1\underline{p}\,[{}^0_1M] = {}^2\underline{p}\,[{}^0_2M]$$

and

$${}^1\underline{n}\,[{}^0_1M] = \pm\,{}^2\underline{n}\,[{}^0_2M]$$

where

${}^1\underline{p}$ - Point P with reference to S_1

${}^2\underline{p}$ - Point P with reference to S_2

${}^1\underline{n}$ - Unit normal vector of the surface Σ_1 at point P with reference to S_1

${}^2\underline{n}$ - Unit normal vector of the surface Σ_2 with respect to S_2

$[{}^0_1M]$ - Homogeneous transformation matrix transforming a vector from the coordinate system S_1 to S_0

$[{}^0_2M]$ - Homogeneous transformation matrix transforming a vector from the coordinate system S_2 to S_0

It is to be noted that all vectors such as position vectors, normal vectors and velocity vectors are all expressed as row vectors in the homogeneous coordinate system. Hence the fourth element of any position vector is unity and it is zero of the normal vectors.

The surface Σ_2 is the boundary of the swept volume of the cutter surface Σ_1 which is nothing but the envelope to the various instances of the cutter surface. In order to identify the point on the cutter surface Σ_1 which is cutting the blank and producing the surface Σ_2 at any given time, it is necessary to have some additional constraint called the condition of contact. It can be said that, in order to ensure a smooth contact or cutting at given time time, the relative velocity vector between the cutter and the blank must be orthogonal to the common normal at the point of cutting [23]. Mathematically, it can be expressed as

$$ {}^{0}\underset{\sim}{n}_{p,1} \cdot {}^{0}\underset{\sim}{V}_{p,12} = 0 $$

and

$$ {}^{0}\underset{\sim}{n}_{p,2} \cdot {}^{0}\underset{\sim}{V}_{p,12} = 0 $$

where

${}^{0}\underset{\sim}{n}_{p,1}$ - Unit normal vector of the cutter surface Σ_1 at point P with reference to the coordinate system S_0

${}^{0}\underset{\sim}{n}_{p,2}$ - Unit normal vector of the cutter surface Σ_2 at point P with reference to the coordinate system S_0

${}^{0}\underset{\sim}{V}_{p,12}$ - Relative velocity vector between the tool and blank with reference to the coordinate system S_0

The condition of contact is a scalar equation with parameters u, v and t. Thus the condition of contact is a relation between u and v of any given instance of time. For any value within its range , a corresponding value of the parameter v can be found. Thus one can locate a set of points or a curve on the surface which defines the surface curve for Σ_2 as well. The corresponding points of the surface can be obtained using the equation

$$ {}^{2}\underset{\sim}{p} = {}^{1}\underset{\sim}{p} \left[{}^{0}_{1}M\right] \left[{}^{2}_{0}M\right] $$

Formally the symbolic algorithm can be stated as follows:

2.3 THE SYMBOLIC ALGORITHM:

1. Define the position vector ${}^{2}\underset{\sim}{p}$ of a generic point P on the cutter surface as a biparametric surface in u and v such that

$${}^{1}\underset{\sim}{p}(u,v) = [x_1(u,v) \;\; y_1(u,v) \;\; z_1(u,v) \;\; 1]$$

item Define the twelve parameters of motion $(a_i, b_i, c_i, \theta_i, \phi_i, \psi_i), i = 1, 2$ in terms of time t.

2. Calculate the matrices $\left[{}^{0}_{1}M\right]$ and $\left[{}^{0}_{1}\dot{M}\right]$ as follows:

$$\left[{}^{0}_{1}M\right] = \left[{}^{0}_{1}R_x\right] \left[{}^{0}_{1}R_y\right] \left[{}^{0}_{1}R_z\right] \left[{}^{0}_{1}T\right]$$

where

$$\left[{}^{0}_{1}R_x\right] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_1 & \sin\theta_1 & 0 \\ 0 & -\sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \left[{}^{0}_{1}R_y\right] = \begin{bmatrix} \cos\phi_1 & 0 & -\sin\phi_1 & 0 \\ 0 & 1 & 0 & 0 \\ \sin\phi_1 & 0 & \cos\phi_1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\left[{}^{0}_{1}R_z\right] = \begin{bmatrix} \cos\psi_1 & sin\psi_1 & 0 & 0 \\ -\sin\psi_1 & \cos\psi_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \left[{}^{0}_{1}T\right] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ a_1 & b_1 & c_1 & 1 \end{bmatrix}$$

3. Calculate the matrices $\left[{}^{0}_{2}M\right], \left[{}^{2}_{0}M\right]$ and $\left[{}^{0}_{2}\dot{M}\right]$ as follows:

$$\left[{}^{0}_{2}M\right] = \left[{}^{0}_{2}R_x\right] \left[{}^{0}_{2}R_y\right] \left[{}^{0}_{2}R_z\right] \left[{}^{0}_{2}T\right]$$

where

$$\left[{}^{0}_{2}R_x\right] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_2 & \sin\theta_2 & 0 \\ 0 & -\sin\theta_2 & \cos\theta_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \left[{}^{0}_{2}R_y\right] = \begin{bmatrix} \cos\phi_2 & 0 & -\sin\phi_2 & 0 \\ 0 & 1 & 0 & 0 \\ \sin\phi_2 & 0 & \cos\phi_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\left[{}^0_2R_z\right] = \begin{bmatrix} \cos\psi_2 & sin\psi_2 & 0 & 0 \\ -\sin\psi_2 & \cos\psi_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \left[{}^0_2T\right] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ a_2 & b_2 & c_2 & 1 \end{bmatrix}$$

$$\left[{}^2_0M\right] = \left[{}^0_2M\right]^{-1}$$

Matrix $\left[{}^0_2\dot{M}\right]$ is the time-differential of matrix $[{}^0_2M]$

4. Derive the position vector

$${}^2\underset{\sim}{p}(u,v) = [x_2(u,v) \;\; y_2(u,v) \;\; z_2(u,v) \;\; 1]$$

using

$${}^2\underset{\sim}{p} = {}^1\underset{\sim}{p}\left[{}^0_1M\right]\left[{}^2_0M\right]$$

5. Derive the normal vector

$${}^1\underset{\sim}{N}_{p,1} = \frac{\partial\,{}^1\underset{-}{p}}{\partial u} X \frac{\partial\,{}^1\underset{-}{p}}{\partial v}$$

where

$\frac{\partial\,{}^1\underset{-}{p}}{\partial u}$ and $\frac{\partial\,{}^1\underset{-}{p}}{\partial v}$ are the partial derivatives of ${}^1\underset{-}{p}$ with respect to u and v respectively so that

$$\frac{\partial\,{}^1\underset{-}{p}}{\partial u} = \left[\frac{\partial x_1}{\partial u} \;\; \frac{\partial y_1}{\partial u} \;\; \frac{\partial z_1}{\partial u} \;\; 0\right]$$

$$\frac{\partial\,{}^1\underset{-}{p}}{\partial v} = \left[\frac{\partial x_1}{\partial v} \;\; \frac{\partial y_1}{\partial v} \;\; \frac{\partial z_1}{\partial v} \;\; 0\right]$$

and hence

$${}^1\underset{\sim}{N}_{p,1} = \begin{vmatrix} i & j & k \\ \frac{\partial x_1}{\partial u} & \frac{\partial y_1}{\partial u} & \frac{\partial z_1}{\partial u} \\ \frac{\partial x_1}{\partial v} & \frac{\partial y_1}{\partial v} & \frac{\partial z_1}{\partial v} \end{vmatrix}$$

6. Calculate the unit normal vector

$$ {}^{1}\underset{\sim}{n}_{p,1} = \frac{{}^{1}\underset{-}{N}_{p,1}}{||{}^{1}\underset{-}{N}_{p,1}||} $$

where $||{}^{1}\underset{-}{N}_{p,1}||$ is the norm of ${}^{1}\underset{-}{N}_{p,1}$

7. Calculate the unit normal vector ${}^{0}\underset{-}{n}_{p,1}$ in the global coordinate system using

$$ {}^{0}\underset{\sim}{n}_{p,1} = {}^{1}\underset{\sim}{n}_{p,1} \left[{}^{0}_{1}M\right] $$

8. Calculate the relative velocity vector

$$ {}^{0}\underset{\sim}{V}_{p,12} = {}^{1}\underset{\sim}{v}_{p,2} - {}^{1}\underset{\sim}{V}_{p,1} $$

where

$$ {}^{1}\underset{\sim}{V}_{p,1} = {}^{1}\underset{\sim}{p} \left[{}^{0}_{1}\dot{M}\right] $$

and

$$ {}^{1}\underset{\sim}{V}_{p,2} = {}^{1}\underset{\sim}{p} \left[{}^{0}_{2}\dot{M}\right] $$

9. Reduce symbolically the condition of contact

$$ {}^{0}\underset{\sim}{n}_{p,1} . {}^{0}\underset{\sim}{V}_{p,12} = 0 $$

10. ${}^{2}\underset{-}{p}$ obtained in step 5 will be a function of u, v and t. Use the condition of contact obtained in step 10 to eliminate either of u and v. This will yield Σ_2

2.4 EXAMPLES

In this section two examples of the cams are discussed along with the implementation of the symbolic algorithm. Both of these are implemented using starbase graphics[47] using double buffering and are shown in animation.

2.4.1 TWO DIMENSIONAL CAM:

The first example is a two dimensional disc cam with a translating flat-faced follower (Fig.2-3). The flat face is perpendicular to the direction of translatory motion of the follower; a is the offset distance ana r is the radius of the circular cam [23].

The cam rotates with the uniform angular velocity t. The angular rotation ψ_1 is the parameter of the cam motion. The follower's linear displacement k is measured from O_0.

The twelve parameters of motion are

$$\begin{array}{cc} a_1 = k\ sint & a_2 = 0 \\ b_1 = -k\ cost & b_2 = r - k\ cost \\ c_1 = 0 & c_2 = 0 \\ \theta_1 = 0 & \theta_2 = 0 \\ \phi_1 = 0 & \phi_2 = 0 \\ \psi_1 = t & \psi_2 = 0 \end{array}$$

and

$$^1\underset{\sim}{p} = [r\ cosu \quad r\ sinu \quad v \quad 1]$$

Proceeding as discussed in the section 2.2 above,

$$^2\underset{\sim}{p} = [k\ sint + r\ cos(u+t) \quad -r + r\ sin(u+t) \quad v \quad 1]$$

with the condition of contact

$$u + t = \frac{\pi}{2}.$$

Eliminating t, we get

$$x_2 = \pm k \ cosu$$

$$y_2 = 0$$

$$z_2 = v$$

As the discussion is about a two dimensional cam, putting $z = 0$, the corresponding curve is given by

$$x_2 = \pm k \ cosu$$

$$y_2 = 0$$

$$z_2 = 0$$

which is the equation of Σ_2.

This is implemented in starbase and the cam in motion in graphic display. A particular instance is shown in Fig.2-4.

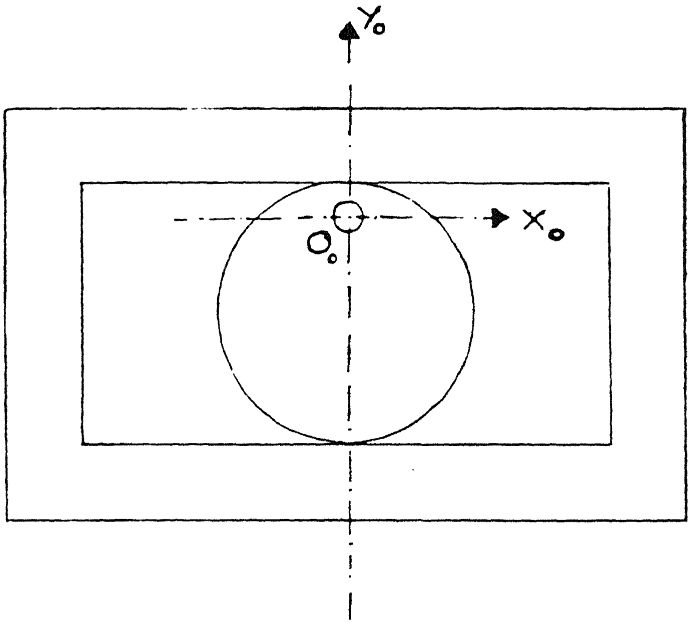

Figure 2-3 Two dimentional Cam - Translating flat faced follower

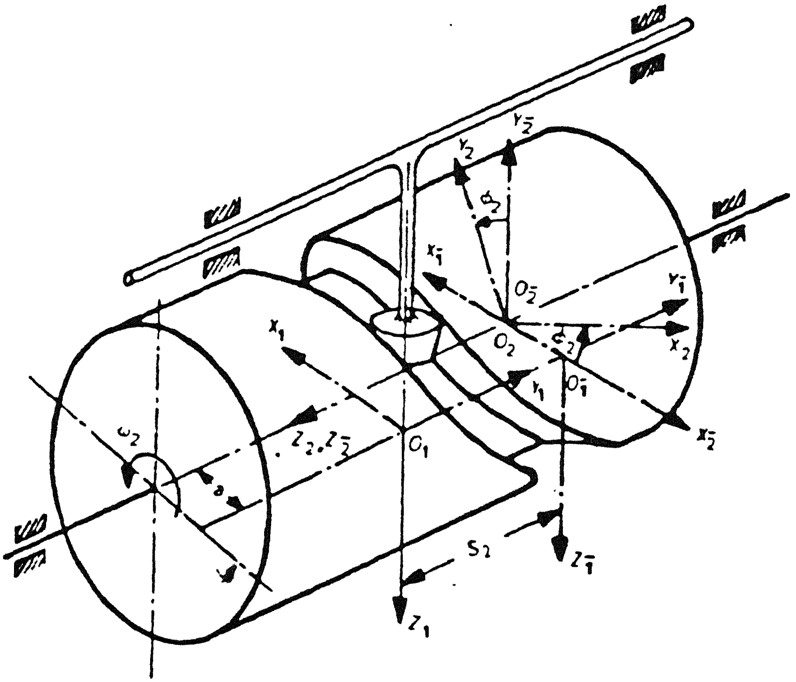

Figure 2-4 Cylindrical Cam - Translating follower

2.4.2 CYLINDRICAL CAM - TRANSLATING FOLLOWER:

The second example considered is a cylindrical cam with translating offset follower as shown in Fig 2-4. Here a is the amount of offset and the parameters are follows:

$$\begin{array}{cc} a_1 = -a & a_2 = 0 \\ b_1 = 0 & b_2 = 0 \\ c_1 = \frac{s}{2}\, sint & c_2 = 0 \\ \theta_1 = \frac{\pi}{2} & \theta_2 = 0 \\ \phi_1 = 0 & \phi_2 = 0 \\ \psi_1 = 0 & \psi_2 = t \end{array}$$

and

$${}^{1}\underset{\sim}{p} = [r\ cosu \quad r\ sinu \quad v \quad 1]$$

Thus

$$x_2 = r\ cosu\ cost - v\ sint - a\ cost$$

$$y_2 = -r\ cosu\ sint - v\ cost + a\ sint$$

$$z_2 = r\ sinu + \frac{s}{2}\ sint$$

with the condition of contact

$$v = \frac{s}{2}\ cost\ tanu$$

.

Eliminating t from the two equations (2.1) and (2.2) the surface Σ_2 can be obtained.

2.5 ANIMATION AND GRAPHICS DISPLAY:

The graphics display of the first example is fairly straight forward as the surfaces Σ_1 and Σ_2 are a circular disc and two parallel lines touching the circle.(Fig.2-4). However, the display of the cylindrical cam with translating follower is a bit tricky. The symbolic algorithm generates only the surface Σ_2 which is conjugate to the cutter surface. To have the realism in display and animation, it is necessary to draw the cylindrical cam and the cutter along with the surface generated in motion. This requires finding the intersection curves of the cylinder with the surface Σ_2. This results in the following discussion:

Let R be the radius of the cylinder with it's angular velocity t. Then the equation of the cylinder is

$$x = R\,cost$$

$$y = R\,sint$$

$$z = d$$

where d is the height of the cylinder(see Fig 2-6).

The surface calculated from the symbolic algorithm, given in equation(2.1) and (2.2) is

$$x_2 = r\ cosu\ cost - v\ sint - a\ cost$$

$$y_2 = -r\ cosu\ sint - v\ cost + a\ sint$$

$$z_2 = r\ sinu + \frac{s}{2}\ sint$$

with

$$v = \frac{s}{2}\ cost\ tanu.$$

The intersection curve of the cylinder and the surface should be a circle in the XY-Plane (see Fig.2-5). Hence,

$$x^2 + y^2 = x_2^2 + y_2^2.$$

Simplifying and eliminating v, we get

$$4\ r^2\ cos^4u - 8\ a\ r\ cos^3u + (4\ a^2 - 4\ R^2 - s^2\ cos^2t)\ cos^2u + s^2\ cos^2t = 0.$$

This is a fourth degree polynomial equation in *cosu* for each instance of t and hence gives rise to four roots for *cosu*. However, the roots are meaningful only if they lie in the interval $[-1,1]$.

Analyzing further mathematically with the constraint that $|cosu| \leq 1$, it can be seen that only two valid roots for *cosu* are possible for each instance of t. These two roots give rise to two curves which define the boundary of the surface Σ_2 as shown in Fig.2-5.

Once the curves are found then the following algorithm illustrates the display and the animation of the cylindrical cam along with the translating follower. Some instances of these are shown in Figures 2-6,2-7 and 2-8.

2.6 ALGORITHM:

1. Define the cutter surface

$$[r\ cosu \quad r\ sinu \quad v \quad 1] \qquad 0 \leq v \leq d$$

2. For t = 0 to MAXT {

a) Define the cylinder

$$x = R\ cosu$$

$$y = R\ sinu$$

$$z = f_1(t)$$

where $f_1(t)$ is one of the two intersection curves nearer to z = 0

b) Define the cylinder

$$x = R\ cosu$$

$$y = R\ sinu$$

$$z = f_2(t)$$

where $f_2(t)$ is the intersection curve nearer to z = d.

c) Define the cylinder

$$x = R\ cosu$$

$$y = R\ sinu$$

$$z = v \qquad f_1(t) \leq v \leq f_2(t)$$

d) Define the circular discs

$$x = R_1\ cosu$$

$$y = R_1\ sinu$$

$$z = f_1(t)$$

and

$$x = R_1\ cosu$$

$$y = R_1\ sinu$$

$$z = f_2(t)$$

}

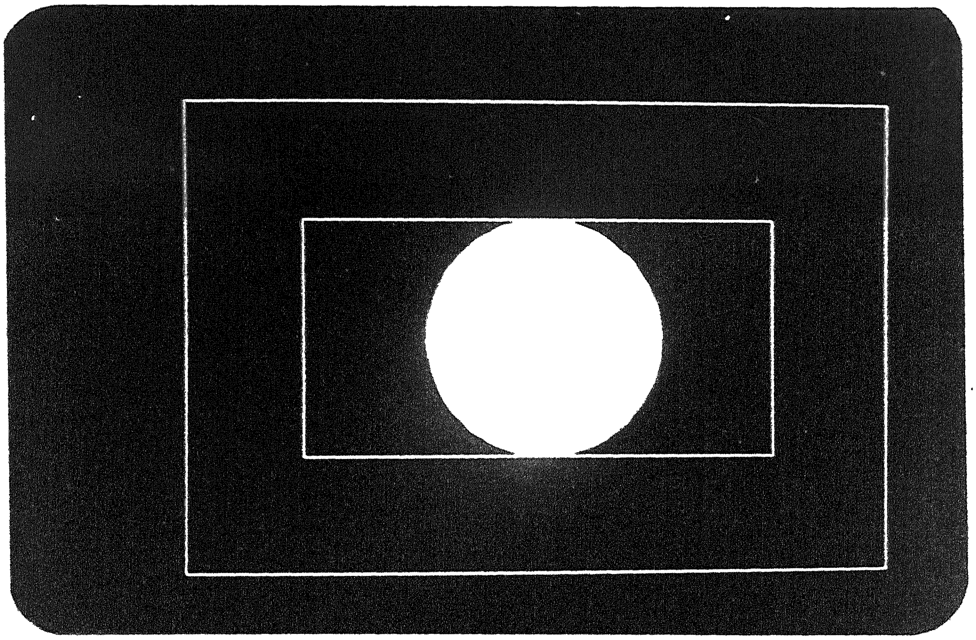

Figure 2-5 Animation of Two dimentional Cam - flat faced follower

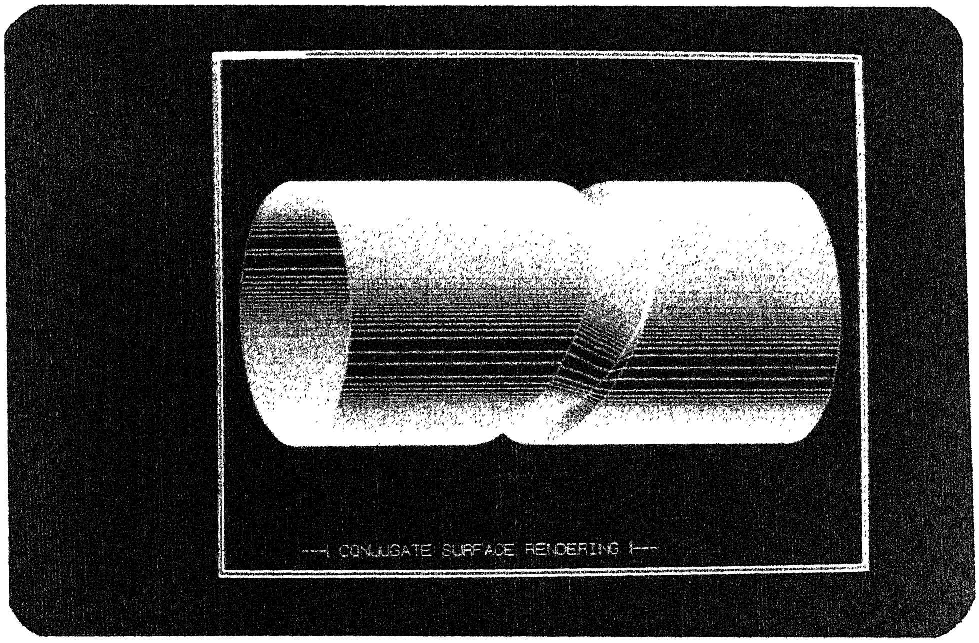

Figure 2-6 Conjugate Surface - Three dimentional Cam

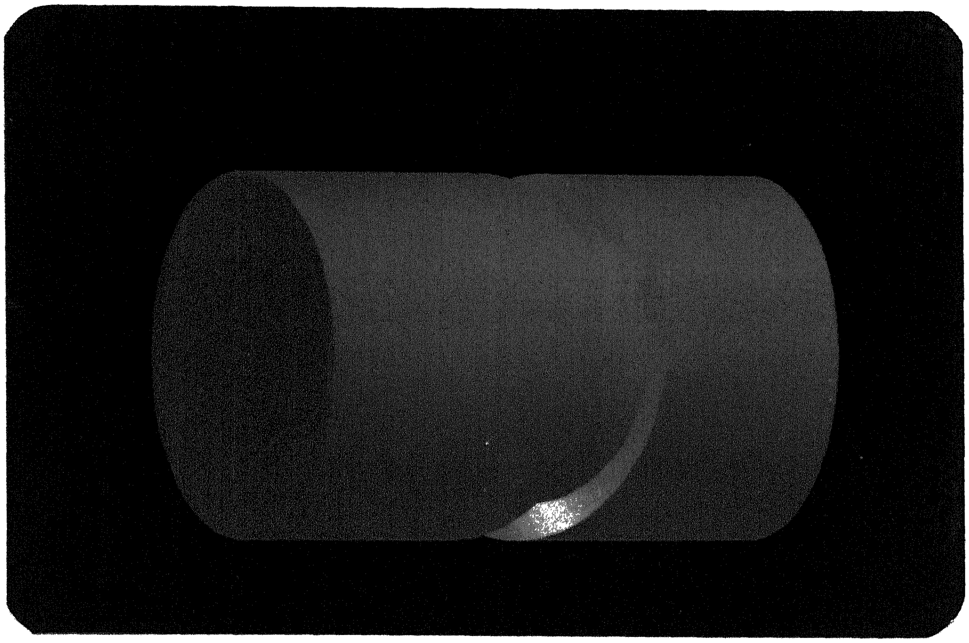

Figure 2-7 Animation of Conjugate Surface - Three Dimensional Cam

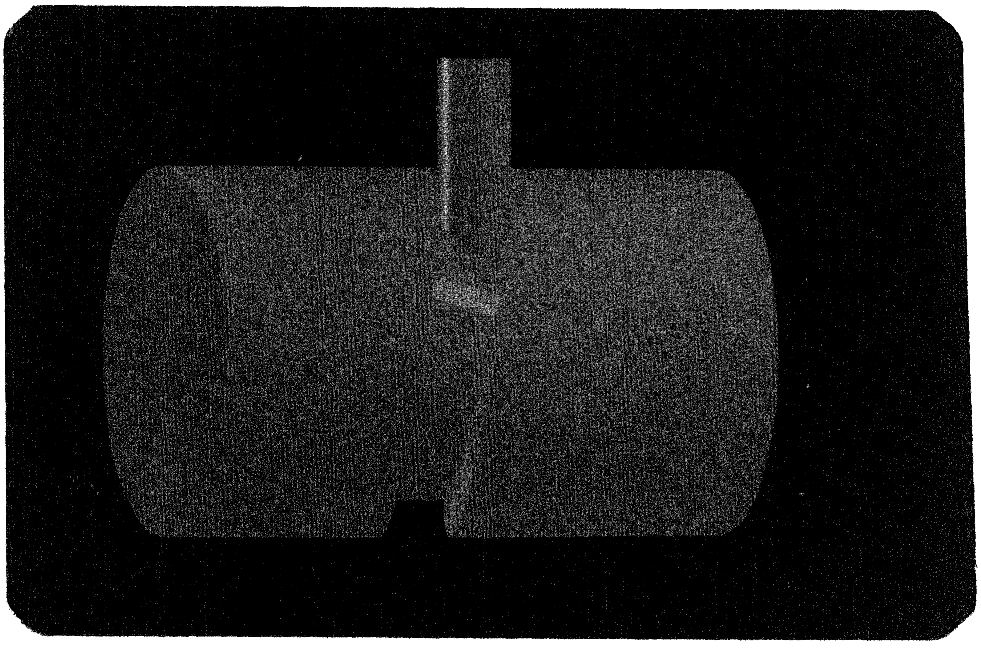

Figure 2-8 Animation of Three Dimensional Cam and follower

Chapter 3

COMPUTER ANIMATION OF CONJUGATE SURFACES

3.1 INTRODUCTION:

In this chapter, the algorithms for generating the cutter surface, it's conjugate surfaces and the algorithms used for animation of conjugate surfaces are presented. A model of the cutter determined by nine parameters [27] is used to generate the cutter. Another twenty four parameters determine the initial and final positions of the cutter. These parameters are the twelve parameters of the symbolic algorithm for the initial position of the cutter and the twelve parameters for the final position of the cutter.

The motion of the cutter or the in-betweening is simulated through the technique of linear interpolation between the initial and final parameters of the cutter. The details are discussed in subsequent sections.

3.2 BASIC CONCEPTS:

The symbolic algorithm coupled with the technique of in-betweening using linear interpolation is the basic methodology used for producing the animation . Before going into the details, it is necessary to describe the definition and the algorithm to generate the generic cutter.

3.2.1 DEFINITION OF A GENERIC CUTTER:

The geometry of a cutter can be modeled as an axisymmetric, biparametric surface. Dhande and Karunakaran [27] proposed a model of the generic cutter in which the generatrix curve is a planar curve consisting of a set of contiguous straight line segments and circular fillet segments. The directrix motion is the axisymmetric sweep of this curve around an axis in the plane of the generatrix curve. The generatrix curve is assumed to be a piecewise continuous curve consisting of three straight line segments and a circular fillet (see Fig.3-1). This generatrix curve can be defined by means of eight parameters, namely, d, r, e, f, a, b, h and h_1 as shown in Fig.3-1. The cutter surface can be obtained by rotating the generatrix curve along the z-axis.

3.2.2. DESCRIPTION OF THE GENERIC CUTTER:

The planar curve of the generic cutter surface consists of three straight line segments and a circular arc segment(Fig.3-1). A coordinate system $S_c(O_c - X_c, Y_c, Z_c)$ is considered to be attached to this cutter profile. The planar curve of a cutter profile can be represented by means of a parameter u which is the distance of any generic point on the profile with respect to the origin O_c. The cutter surface Σ_1 is obtained by sweeping this generatrix profile along Z_c axis. The four segments of the generatrix curve can be algebraically expressed as

$$O_cP : x_c\ tana$$

$$PQ : (x_c - e)^2 + (z_c - f)^2 = r^2$$

$$QS : x_c = z_c\ tanb - \frac{d}{2}\ tana\ tanb + \frac{d}{2}$$

$$ST : x_c = h\ tanb - \frac{d}{2}\ tana\ tanb + \frac{d}{2}$$

where

O_cP - First straight line segment

PQ - The circular segment with center at A

QS - Second straight line segment

ST - Third straight line segment

R - Point of intersection the first segment O_cP and the third segment QS.

d - Diameter of the cutter measured at R

r - Radius of the circular segment. This is positive for convex arc and negative for concave arc.

a - Angle of segment O_cP measured from X_c axis $(-\frac{\pi}{2} < a < \frac{\pi}{2})$.

b - Angle of segment QS measured from Z_c axis $(-\frac{\pi}{2} < a < \frac{\pi}{2})$

e - X coordinate of point A in the coordinate system S_c

f - Z coordinate of point A in the coordinate system S_c

h - Non parallel height of the cutter

h_1 - Parallel height of the cutter

is-fillet-radius - Flag set to TRUE if O_cP and QS are tangents to the circular segment PQ else FALSE.

u - Parameter which defines the distance of any generic point P on the profile measured from O_c

v - Parameter describing the sweep of the two dimensional profile

$[{}^1_cM]$ - Homogeneous transformation matrix describing the sweep of the two dimensional profile

s_1, s_2, s_3, s_4 - lengths of the segments O_cP, PQ, QS and ST respectively

${}^c\underset{\sim}{p}$ - position vector of a generic point P on the two dimensional profile of the cutter in S_c coordinate system in terms of u

${}^1\underset{\sim}{p}$ - Position vector of a generic point P on the surface of the cutter in S_1 coordinate system in terms of u and v

The parametric equation of the generatrix curve ${}^c\underset{\sim}{p}(u)$ with respect to the coordinate system $S_c(O_c \; - \; X_c, Y_c, Z_c)$ is defined by

$$ {}^c\underset{\sim}{p}(u) = [x_c(u) \quad y_c(u) \quad z_c(u) \quad 1] $$

where

$$x_c(u) = \begin{cases} u\ cosa, & 0 \leq u < s_1 \\ e + |r|\ cos\{\arctan(\frac{s_1\ sina - f}{s_1\ cosa - e}) + (\frac{u - s_1}{r})\}, & s_1 < u \leq s_1 + s_2 \\ u - (s_1 + s_2 + s_3)\ sinb + x_c^s, & s_1 + s_2 < u \leq s_1 + s_2 + s_3 \\ x_c^s, & s_1 + s_2 + s_3 < u \leq s_1 + s_2 + s_3 + \end{cases}$$

$$y_c(u) = 0$$

$$z_c(u) = \begin{cases} u\ sina, & 0 \leq u < s_1 \\ f + |r|\ sin\{\arctan(\frac{s_1\ sina - f}{s_1\ cosa - e}) + (\frac{u - s_1}{r})\}, & s_1 < u \leq s_1 + s_2 \\ u - (s_1 + s_2 + s_3)\ cosb + z_c^s, & s_1 + s_2 < u \leq s_1 + s_2 + s_3 \\ u - (s_1 + s_2 + s_3) + z_c^s, & s_1 + s_2 + s_3 < u \leq s_1 + s_2 + s_3 + \end{cases}$$

Then the cutter surface Σ_1 is given by

$$^{1}\underset{\sim}{p}(u,v) = {}^{c}\underset{\sim}{p}(u)[^{1}_{c}M]$$

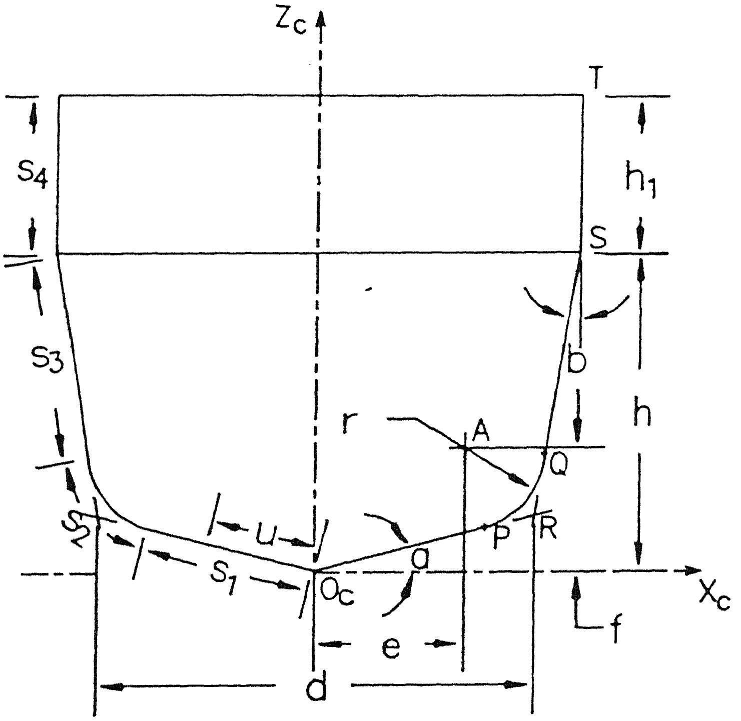

Figure 3-1. Definition of a Generic Cutter

3.2.3 ALGORITHM:

In this algorithm x_c^J, y_c^J denote the x, y coordinate values of the point J.

1. Input the values of is-fillet-radius, d, r, a, b, h_1

2. If(is-fillet-radius){

$$e = \frac{d}{2} - r(\frac{(cosa - sinb)}{cos(a+b)})$$

$$f = \frac{r + e\ sina}{cosa}$$

   }

   else input the values of e and f

3. Calculate the point P

$$x_c^P = (e + f\ tana)cos^2 a - \sqrt{[(e + f\ tana)cos^2 a]^2 - (e^2 + f^2 - r^2)cos^2 a}$$

$$z_c^p = x_c^P\ tana$$

4. Calculate the point Q

$$z_c^Q = \{f + [e + \frac{d}{2}(tana\ tanb - 1)]tanb\}cos^2 b$$
$$+\sqrt{\begin{array}{l}[\{f + [e + \frac{d}{2}(tana\ tanb - 1)]tanb\}cos^2 b]^2 \\ -\{f^2 + [e + \frac{d}{2}(tana\ tanb - 1)]^2 - r^2\}cos^2 b\end{array}}$$

$$x_c^Q = z_c^Q\ tanb - \frac{d}{2}(tana\ tanb - 1)$$

5. Calculate h

   If(is-fillet-radius) and (b = 0) $h = z_c^Q$

   else input the value of h

6. Calculate the point S

$$x_c^S = h\ tanb - \frac{d}{2}\ tana\ tanb + \frac{d}{2}$$

$$z_c^S = h$$

7. Calculate s_1, s_2, s_3 and s_4

$$s_1 = \sqrt{(x_c^P)^2 + (z_c^P)^2}$$

$$s_2 = r\ \{\arctan(\frac{z_c^Q - f}{x_c^Q - e}) - \arctan(\frac{z_c^P - f}{x_c^P - e})\}$$

$$s_3 = \sqrt{(x_c^S - x_c^Q)^2 + (Z_c^S - Z_c^Q)^2}$$

$$s_4 = h_1$$

8. Define the parametric equation ${}^{c}\underset{\sim}{p}(u)$ of the two dimensional profile of the cutter in terms of the parameter u

$${}^{c}\underset{\sim}{p}(u) = [x_c(u)\quad y_c(u)\quad z_c(u)\quad 1]$$

as follows:

$$y_c(u) = 0, s_1 \leq u \leq s_4$$

If$(0 \leq u < s_1)$\{

$$x_c(u) = u\ cosa$$

$$z_c(u) = u\ sina$$

\} else if$(s_1 \leq u < s_1 + s_2)$\{

$$x_c(u) = e + |r|\ cos\{\arctan(\frac{s_1\ sina - f}{s_1\ cosa - e}) + (\frac{u - s_1}{r})\}$$

$$z_c(u) = f + |r|\ sin\{\arctan(\frac{s_1\ sina - f}{s_1\ cosa - e}) + (\frac{u - s_1}{r})\}$$

\}

```
else if($s_1 + s_2 \le u < s_1 + s_2 + s_3$){
        $x_c(u) = u - (s_1 + s_2 + s_3)\ sinb + x_c^s$
        $z_c(u) = u - (s_1 + s_2 + s_3)\ cosb + z_c^s$
}
else if($s_1 + s_2 + s_3 \le u < s_1 + s_2 + s_3 + s_4$){
        $x_c(u) = x_c^s$
        $z_c(u) = u - (s_1 + s_2 + s_3) + z_c^s$
}
```

9. Define the sweep matrix $[^1_cM]$
10. Calculate ${}^1\underset{\sim}{p}$(u,v) using

$$^1\underset{\sim}{p}(u,v) = {}^c\underset{\sim}{p}(u)[^1_cM]$$

3.3 METHODOLOGY:

In the previous chapters and the previous sections, the algorithms for the conjugate surfaces and the shape of the cutter are discussed. However, they involve symbolic computation and hence cannot be implemented directly. A symbolic computation program MACSYMA is used to define this task. This finds the condition of contact and the corresponding x, y and z values. This section describes the algorithm used for generating the conjugate surfaces at each time frame, the trimming surfaces and the algorithm used to show the animation of conjugate surfaces.

Algorithm 3.3.1 explains the in-betweening and the generation of the conjugate surfaces while algorithm 3.3.2 describes the generation of the trimming surfaces and finally algorithm 3.3.3 illustrates the way animation is modeled.

3.3.1 ALGORITHM:

```
1. Input 24 parameters of movement $a_{ij}$, $b_{ij}$, $c_{ij}$, $\theta_{ij}$, $\phi_{ij}$, $\psi_{ij}$   $i, j = 1, 2$

2. Input the 9 parameters of the cutter $is-fillet-radius, a, b, d, e, f, h, h_1, r$

3. Calculate $s_1, s_2, s_3, s_4$ using algorithm 3.2.3

4. For time = 0 to MAXT {
    t = time/MAXT
    for i = 1,2 {
                          $a_i = a_{i1} + (a_{i2} - a_{i1})t$
                          $b_i = b_{i1} + (b_{i2} - b_{i1})t$
                          $c_i = c_{i1} + (c_{i2} - c_{i1})t$
                          $\theta_i = \theta_{i1} + (\theta_{i2} - \theta_{i1)}t$
                          $\phi_i = \phi_{i1} + (\phi_{i2} - \phi_{i1})t$
                          $\psi_i = \psi_{i1} + (\psi_{i2} - \psi_{i1})t$
    }
   }

5. For each of the four segments of the cutter, namely,
   $S_i$ to $S_{i+1}$ ,i = 0 to 3
   calculate $x_2(u,v), y_2(u,v), z_2(u,v), con(u,v)$ using the algorithm 2.3

6. For $i = 0$ to 3{
   for u = $s_i$ to $s_{i+1}$ do {
  a ) Calculate the values of v which satisfy the condition con(u,v) = 0
  b ) For each v in (a) do {
  1 . calculate
                              $x = x_2(u,v)$
                              $y = y_2(u,v)$
```

```
                    $z = z_2(u,v)$
    2 . Draw the surface with values x, y, z
              }
          }
      }
```

The algorithm needs some some explanation. The in-betweening is done using linear interpolation of the 24 parameters of movement. Out of these, twelve parameters corresponding to the initial position and the remaining twelve correspond to the final position. These are linearly interpolated in time. This is ensured in step 4. In step 5, $x_2(u,v), y_2(u,v), z_2(u,v)$ and $con(u,v)$ are calculated symbolically using the algorithm 2.3 . Here symbolic computation program MACSYMA is used to generate $x_2(u,v), y_2(u,v), z_2(u,v)$ and $con(u,v)$ in symbolic form.

In reality, either all values of v satisfy the condition of contact $con(u,v)$ in the range $[0,2\pi]$ or there will be exactly two values of v satisfying the condition of contact in the range $[0,2\pi]$. Moreover, these two values of v will be almost π apart. This is because the cutter shape is assumed to be axisymmetric. These values of v can be determined by a numerical method. (In this thesis regula-falsi method is used). These give rise to eight different surfaces.

Once the surfaces are drawn, the next step would be to trim the ends of the surfaces. This can be accomplished by drawing two more surfaces at the initial and final positions. The following algorithm illustrates the methodology to draw the trimming surfaces. An instance of the initial trimming surface and an instance of final trimming surface are shown in Fig.3-3.

3.3.2 ALGORITHM:

```
1.Put t = 0 in algorithm 3.3.1
2. For each u in step 6 of the algorithm 3.3.1 do
3. If there are not more than two roots in $[0,2\pi]$ for con(u,v) = 0 then
```

```
a) set $v_1$ = root nearer to 0
b) set $v_2$ = root nearer to $\pi$
4. For $v = v_1$ to $v_2$ do {
a) Calculate x, y and z using algorithm 3.2.3
b) set $x = x\ cosv$
set $z = z\ sinv$
5. Plot x, y, z
            }
        }
}
```

6. Put t = MAXT in step 1 and repeat the steps 2 to 5 with v = v_2 to $vi + 2\pi$ in step 4.

The last step is to get the trimming surface at the final position of the cutter. The first five steps give the initial trimming surface. It is seen that there will be either exactly two values of v or all values of v satisfying the condition of contact in the range $[0, 2\pi]$. The trimming is not necessary when all the values of v satisfy the condition of contact. In other words, trimming needs to be done when there are only two values of v satisfying the condition of contact. Let the values be v_1 and v_2, v_1 be the root nearer to 0 and v_2 be the root nearer to π. The cutter shape being axisymmetric, the points need to be generated for the surface are nothing but the circular sweep from v_1 to v_2. The initial trimming surface is obtained when t = 0 and the final trimming surface is obtained when t = MAXT with the circular sweep being taken from v_2 to $v_1 + 2\pi$.

These algorithms provide the basis for drawing the surfaces. Now the following algorithm illustrates the displaying of the surfaces along with the cutter, in general, for each time frame. Some instances of the animation are shown in Figures 3-4 and 3-5.

3.3.3 ALGORITHM:

```
1. Draw the initial trimming surface using the algorithm 3.3.2
2. For t = 0 to MAXT do {
a) Draw the cutter surface using algorithm 3.2.3
b) Draw the conjugate surfaces using algorithm 3.3.1
}
3. Draw the final trimming surface using algorithm 3.3.1
```

These algorithms are implemented in C on HP-9000/834 TURBO SRX 3D graphics workstations based on HPPA RISC CPU rated at 2.02 MFLOPS (14 MIPS) performance with 24 plane 19" color monitor using the starbase graphics library. Smooth and continuous motion is achieved using the technique of double buffering in starbase.

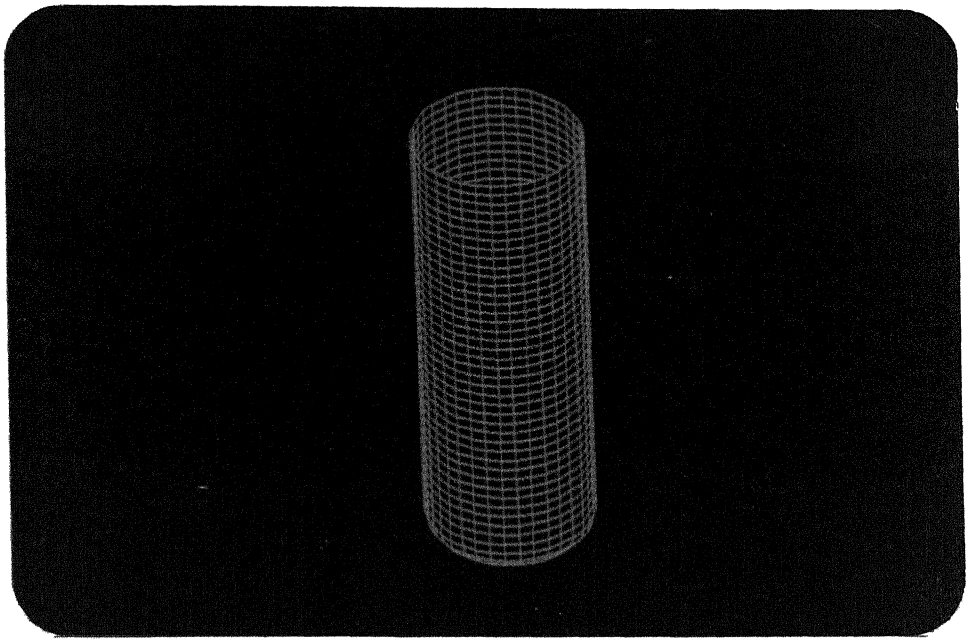

Figure 3-2 A Particular cutter surface

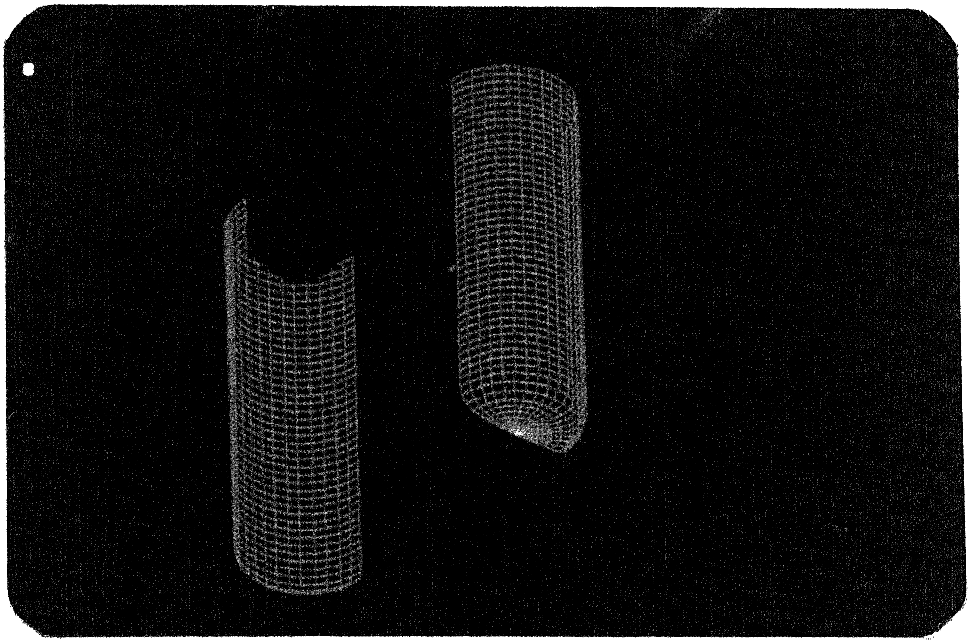

Figure 3-3 Initial and final Trimming surfaces

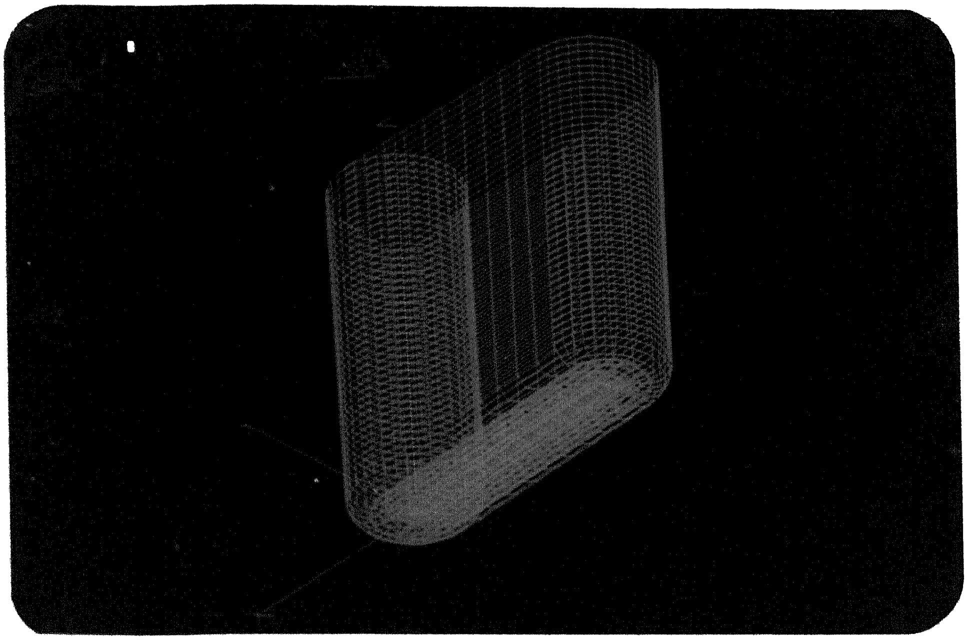

Figure 3-4 Animation of a Conjugate Surface (Translating 50 Units along X-axis)

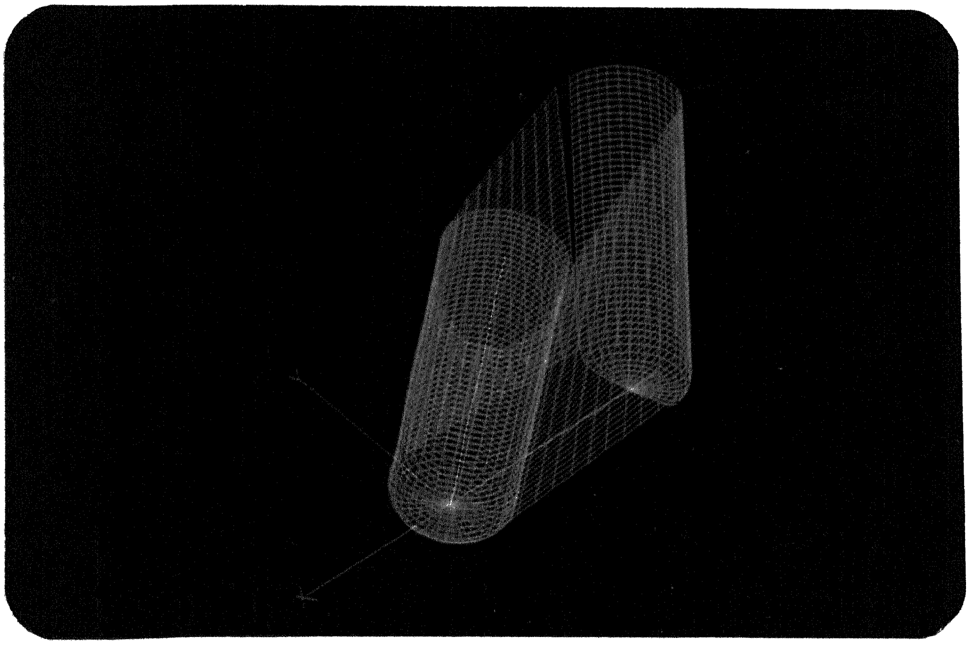

Figure 3-5 Animation of a Conjugate Surface (Translating 50 Units along X-axis and rotating 15° around X-axis)

Chapter 4

SOFTWARE DEVELOPMENT

All the algorithms presented in this thesis are implemented on HP-900/834 Turbo SRX 3D graphics workstation. The total package is divided into six modules (see Fig.4-1). Fig.4-1 illustrates the total methodology adopted for software development.

4.1 CUTTER PROFILE MODULE :

This module takes the nine parameters, namely, is-fillet-radius, a, b, d, e, f, h, h_1 and r. The module generates the two dimensional profile of the cutter according to the input parameters and develops the required input for the next module, namely, surface generation module.

4.2 SYMBOLIC COMPUTATION MODULE:

The symbolic algorithm is implemented using symbolic computation software MACSYMA (see Appendix A). This module requires the twenty four parameters in symbolic form. It generates the expressions for $x_2(u,v,t)$, $y_2(u,v,t)$, $z_2(u,v,t)$ and the condition of contact in terms of u, v and t. The same MACSYMA software may be used to eliminate one of u and v using condition of contact to get the surface Σ_2 in terms of t and one of the parameters u and v. However, there are some inherent properties of the conjugate surfaces which are useful in implementation. For example, for each particular instance of u and t, either all values of v or exactly two values of v will satisfy the condition of contact in the interval $[0, 2\pi]$. Moreover, if there are only two values of v satisfying the condition of contact, then these two values of v differ almost by π. This information is highly valuable in speeding up the symbolic program. This can be achieved by first identifying whether there are exactly two roots and then these two roots can be calculated faster using a numerical method. In this module the method of false position is used to calculate the values of v.

The cutter 2D profile is axisymmetric and it consists of six straight line segments and the two circular segments. Hence, the conjugate surface Σ_2 will be a combination of eight surface segments. Thus, it is sufficient to find the expressions for $x_2(u,v,t)$, $y_2(u,v,t)$, $z_2(u,v,t)$ and the condition of contact for any generic straight line segment and any generic circular segment.

This module produces the expressions for $x_2(u,v,t)$, $y_2(u,v,t)$, $z_2(u,v,t)$ and the condition of contact for any line segment with end points at (l_1, m_1) and (l_2, m_2). Further, it produces the expressions for $x_2(u,v,t)$, $y_2(u,v,t)$, $z_2(u,v,t)$ and the condition of contact for the circular segment, given the center and radius. These will be input to the surface generation module.

4.3 SURFACE GENERATION MODULE:

This module takes the twenty four parameters of motion and also takes the expressions for $x_2(u,v,t)$, $y_2(u,v,t)$, $z_2(u,v,t)$ and the condition of contact in symbolic form. This module calculates the values of v using a numerical method (the method of false position) to eliminate v using the condition of contact. After v is eliminated from $x_2(u,v,t)$, $y_2(u,v,t)$, $z_2(u,v,t)$ using condition of contact, the points x_2, y_2, z_2 are calculated. These x_2, y_2, z_2 account for the eight surfaces which form the conjugate surface Σ_2. These data files (generated by this module) will be input for the trimming algorithm as well as the animation module. In fact, this module is the implementation of the algorithm 3.3.1.

4.4 CUTTER SURFACE GENERATION MODULE:

The cutter surface is a circular sweep of the 2D profile of the cutter that is developed by the cutter profile module. Hence, this module takes input from the cutter profile module and generates the cutter surface by taking circular sweep. Moreover, this module displays the cutter surface as a wire frame model and as a solid model.

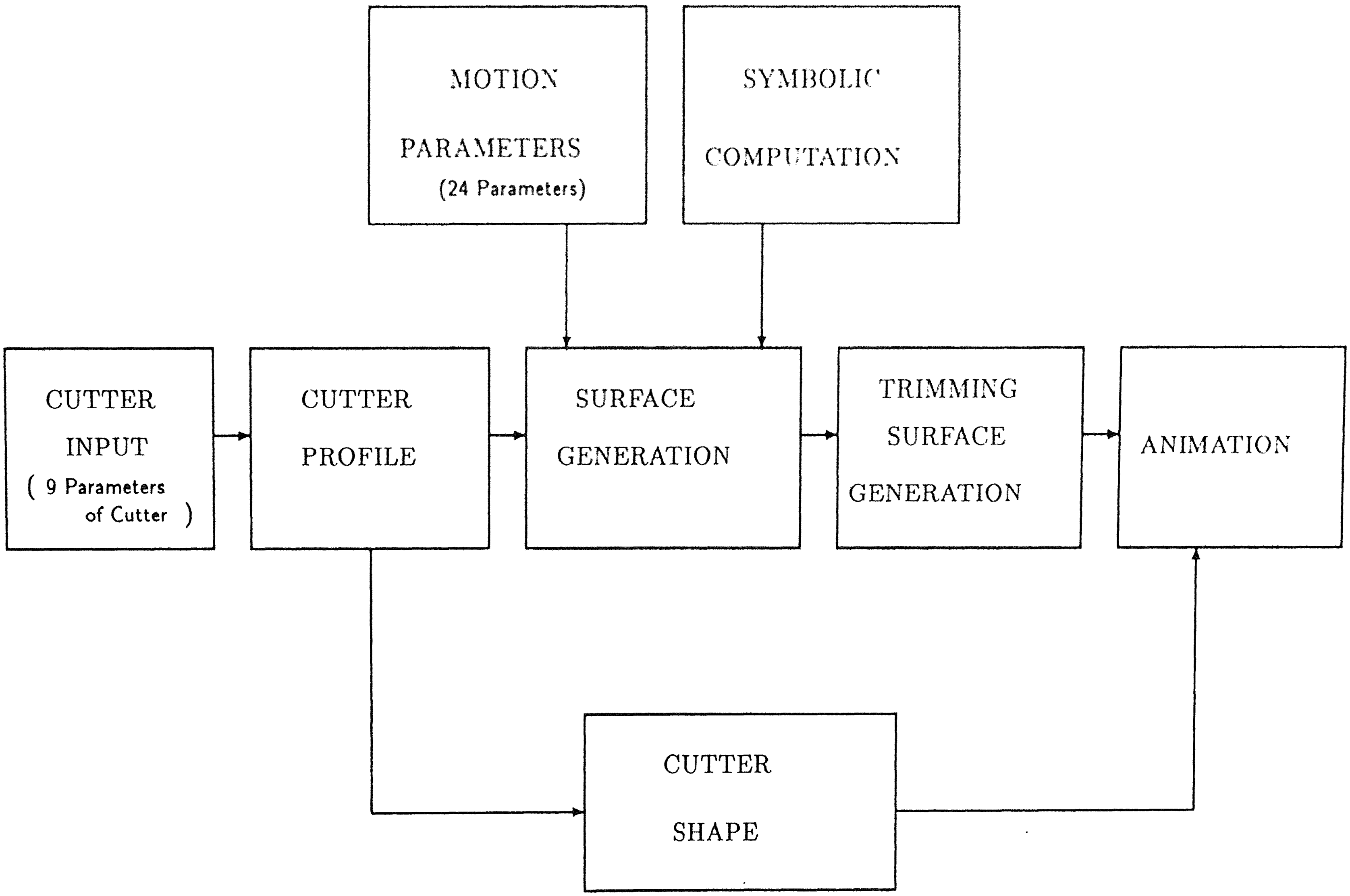

Figure 4- 1 Software Development

4.5 TRIMMING SURFACE GENERATION MODULE:

Trimming is the technique of generating the portion of the cutter surface that will be conjugate to the surface Σ_2 at the initial position of the cutter at t = 0 and at the final position of the cutter at t = MAXT.

This module takes the input from the surface generation module. The input will be the two values of v which satisfy the condition of contact at t = 0 and t = MAXT. Suppose v_1 and v_2 are the values of v satisfying the condition of contact at t = 0 for a particular value of u. A semi-circular sweep from v_1 to v_2 or v_2 to v_1 (depending on the orientation) for all the values of u, gives the trimming surface at the initial position. The trimming surface at the final position of the cutter can be generated in the same way except that the semi-circular sweep is to be taken from v_2 to v_1 (or v_1 to v_2 depending on whether the initial trimming surface is taken from v_1 to v_2 or v_2 to v_1). This module generates the trimming surfaces at the initial and the final positions of the cutter. This module is the implementation of the trimming algorithm 3.3.1.

4.6 ANIMATION MODULE:

The animation module is the implementation of the algorithm 3.3.3. This module draws the cutter surface at the initial position and with each time frame the cutter surface draws the corresponding conjugate surface and finally stops at the final position. The trimming surfaces are also drawn along with the cutter surface.

This module is implemented using the double-buffering technique for movement and the light-source and rendering primitives of starbase graphics library for the rendering and shading the surfaces.

Chapter 5

CONCLUSIONS

The work presented in this thesis outlines a methodology for displaying the conjugate surface generation in animation. A generic cutter with all the twenty four parameters of movement is taken to illustrate the model. Linear interpolation has been used to give the effect of in-betweening in animation.

The symbolic algorithm is used repeatedly to generate the conjugate surfaces and also for the animation purposes, Though, the work shown here is aimed at animating the conjugate surface generation, the algorithms and the methodology proposed in the thesis are general and can be used for modeling any shape generating process.

SCOPE FOR FUTURE WORK:

The in-betweening technique used in this thesis is the linear interpolation. However, in general, various interpolation schemes may be used depending on the application [79]. The interpolation schemes like spline interpolation are used in some applications of animation [51] and they may be used to enhance the speed and reality of this work. The method of quaternions proposed in [33] may improve the algorithms presented in this thesis, which might turn out to be faster than the conventional matrix multiplication methods. Most of the graphics software is developed keeping in view the rendering and other related aspects. However, most of these are not suitable to applications of animation systems. This may be due to the high data requirements of any animation systems. Though there are quite few animation languages[57,61,94], the problem needs to be addressed in much more detail.

Theoretically, if one can specify the class of all surfaces that can be generated using this model then it may be possible for the design engineers to choose the corresponding cutters and shapes. Another important aspect in the theoretical stand point is the limitations of the system and its practical implications.

Appendix A

MACSYMA

Computers have traditionally been used to solve scientific problems that could be expressed in terms of numbers. When a problem can easily be expressed in terms of calculations with numbers, this approach to problem solving works well. On the other hand some problems can be expressed best in symbolic terms or perhaps can only be expressed that way. MACSYMA is a symbolic algebra system. It can work with symbols, polynomial expressions, equations and numbers.

MACSYMA is an interactive symbolic algebra program that offers a wide range of capabilities for solving many kinds of problems in algebra, trigonometry and calculus. The capabilities of MACSYMA include differentiating and integrating expressions, solving equations, manipulating matrices etc. MACSYMA is also a programming environment in which one can define mathematical procedures tailored to his own needs. MACSYMA can return results either in numeric or symbolic form.

CAPABILITIES :

a) The precision in the numerical systems is limited by the systems hardware. MACSYMA can work with exact quantities rather than approximations by carring out the computations in symbolic form. When a symbolic result is converted to a floating-point number, the precision can be set by the user.

b) Constants like π and e can be kept in symbolic form and use them even in floating-point calculations to get exact calculations using them.

c) MACSYMA provides six categories of numbers, namely, INTEGERS, RATIONAL NUMBERS, FLOAT and BIGFLOAT and COMPLEX NUMBERS. MACSYMA does not limit the number of digits in an integer or a rational number but there are some limitations on the non-zero floating point numbers.

d) MACSYMA can be viewed as a programming language. It accommodates many programming structures including conditional statements and loops.

e) MACSYMA provides the facility to convert any expression in symbolic form to the corresponding expression in C or FORTRAN or PASCAL.

Appendix B

STARBASE GRAPHICS LIBRARY

Starbase graphics library is a start of the art package of graphics procedures which, while being based on and supporting graphics standards, provides considerable functionality beyond that defined in graphics standards. The procedures defined in starbase can be accessed from the C, FORTRAN77 and PASCAL programming languages. Starbase is available on many HP-UX systems, including HP 9000 series 300 and 800 computers. Starbase is device independent. This appendix gives a very brief idea of the starbase graphics primitives.

DOUBLE BUFFERING:

Double buffering is a technique for increasing the smoothness of the drawing process on the screen. Double buffering involves taking a multiple-plane frame buffer (24 planes on TSRX systems) and dividing it into two equal parts, Using the **double_buffer** and **dbuffer_switch** procedures, one can ensure that only completed images are displayed. Once an image is complete, it is then displayed. The image in the other half of the frame buffer can then be modified. Toggling back and forth in this manner is called double buffering. Any movement displayed using double buffering will be smooth and avoids jerkyness in movement.

TRANSFORMATION MATRIX STACK:

A transformation matrix performs scaling, rotation, translation etc on the coordinate data. Starbase provides transformation matrices that one can control directly to suit the application . These matrices are arranged in a stack called transformation matrix stack. There are three types of transformations that can modify the stack. They are

a) The modeling transformations

b) The viewing transformations

c) The VDC-to-DC transformations

The modeling transformations are generally useful for the pre-multiplication/post-multiplication of the transformation matrices, pushing on to the stack, poping from the stack etc. some of the generally used routines of modeling transformations are **concat_transformation3d, pop_matrix, push_matrix3d** etc.

The viewing transformations control the synthetic camera, **view-port, view-volume** etc. The routines useful are **view_camera, view_port, view_volume, view_matrix** .

The VDC-to-DC transformations are useful in transforming from virtual device coordinates to device coordinates. The routines useful in these are **mapping_mode, set_ p_1_p_2,vdc_extent, vdc_justification.**

HIDDEN SURFACE REMOVAL :

The starbase uses z-buffer algorithm for hidden surface removal and can be accessed through the primitives **hidden_surface** and **zbuffer_switch.**

RENDERING:

Starbase provides primitives for drawing wire frame models, shading and rendering of the objects. Two primitives **quadrilateral_mesh** and **triangular_mesh** are useful for wire frame models. The light sources and illumination of ambient, specular and reflections can also be modeled using the **shade_mode, light_source** and the **surface_coefficients** routines. Starbase provides for both Phong and Garaud shadings.

Starbase also provides primitives for modeling objects using B-spline curves and surfaces. Some more additional features like panning and zooming, displaying transparency and highlighting are also available in starbase.

BIBILIOGRAPHY

1. Ackland, B and Weste, N., "Real Time Animation Playback on a Frame Store Display System", Proc. SIGGRAPH'80, Computer Graphics, 14(3), 1980, pp 182-188.

2. Badler, N.I., "Human Body Models and Animation", IEEE Computer Graphics and Applications, 2(9), 1982, pp 6-7.

3. Badler, N.I., O'Rourke, J and Kaufman, B., "Special Problems in Human Movement Simulation", Proc. SIGGRAPH'80, Computer Graphics, 14(3), 1980, pp 189-197.

4. Badler, N.I., O'Rourke, J and Toltzis., "A Spherical Representation of a Human Body for Visualizing Movement", Proc. IEEE, vol 67, 10, 1979, pp 1397-1403.

5. Badler, N.I and Smoliar, S.W., "Digital Representations of Human Movement", Computing Surveys, 11(1), 1979, pp 19-38.

6. Baecker, R.M., "Picture Driven Animation", Proc. Spring Joint Computer Conference, AFIPS Press, 34, 1969b, pp 273-288.

7. Boltyanskii, v.c., "Envelopes", MacMillan Company, New York, 1964

8. Baecker, R.M., "Current Issues in Interactive Computer-mediated Animation" Proc. 9th Annual Meeting UAIDE, 1970, pp 273-288.

9. Baecker, R.M., "A Conversational Extensible System for the Animation of Shaded Images", Proc. SIGGRAPH'76.Computer Graphics, 10(2), 1976, pp 32-39.

10. Booth, K.S and Mackay, S., "Techniques for Frame Buffer Animation", Proc. Graphics Interface'82, 1982, pp 213-219.

11. Borrell, J., "The Magic of Computer Animation", Computer Graphics World, 10, 1981, pp 25-33.

12. Burtnyk, N., Pulfer, J.K and Wein M., "Computer Graphics and Film Animation", INFOR, 1971, pp 1-11.

13. Burtnyk, N and Wein M., "Computer-generated Key-frame Animation", Jr. of Society for Motion Picture and Television Engineers, 80, 1971, pp 149-153.

14. Burtnyk, N and Wein M., "A Computer Animation System for the Animator", Proc. UAIDE 10th Annual Meeting, 1971, pp 3-5 to 3-24.

15. Burtnyk, N and Wein M., "Towards a Computer Animating Production Tool", Proc. Eurocomp Congress, Online, Brunel, England, 1974, pp 174-185.

16. Burtnyk, N and Wein M., "Interactive Skeleton Techniques for Enhancing Motion Dynamics in Key Frame Animation", Comm.ACM, Vol 19(10), 1976, pp 564-569.

17. Calvert, T.W and Chapman J., "Notation of Movement With Computer Assistance", Proc. ACM Annual Conf, 2, 1978, pp 731-736.

18. Calvert, T.W and Chapman J., Patla A, "The Integration of Subjective and Objective Data in Animation of Human Movement", Proc. SIGGRAPH'80, Computer Graphics, 14(3), 1980, pp 198-203.

19. Calvert, T.W., Chapman J and Patla A., "Aspects of Kinematic Simulation of Human Movement", IEEE Computer Graphics and Applications 2(9), 1982, pp 41-50.

20. Calvert, T.W., Chapman J and Patla A., "The Simulation of Human Movement", Proc. Graphics Interface'82, 1982, pp 227-234.

21. Catmull E., "The Problems of Computer-assisted Animation", Computer Graphics, 12(3), 1978, pp 348-353.

22. Chuang, R and Entis G., "3D Shaded Computer Animation-step by step", IEEE Computer Graphics and Applications, 3(9), 1983, pp 18-25.

23. Chakraborthy, J and Dhande, S.G., "Kinematics and Geometry of Planar and Spacial Cam Mechanisms", Weily Eastern, New Delhi, 1977.

24. Csuri, C., "Real-time Computer Animation", Proc. IFIP Congress'74, North Holland, 1974, pp 707-711.

25. Csuri, C., "Computer Animation", Proc. SIGGRAPH'75, 1975, pp 92-101.

26. Coquillart, S and Jancéne, P., "Animated Free-Form Deformation: An Interactive Animation Technique", Proc. ACM SIGGRAPH, 1991, pp 27-30

27. Dhande, S.G and Karunakaran, K.P., "Symbolic and Computational Conjugate Geometry of a Generic Cutter and Machine Tool Assembly", Jr. of Design and Manufacture, MIT (communicated)

28. Dhande, S.G and Karunakaran, K.P and Misra, B.K., "Geometric Modeling of Manufacturing Process Using Symbolic and Computational Conjugate Geometry", Jr. of Engg for Industry, Tr. of ASME (communicated).

29. Dietrich F., "A Micro Computer System for Real-time Animation", The Artist Designer and Computer Graphics, Tutorial SIGGRAPH'83, 18, 1983, pp 43-47.

30. Duff, D.S., "Simulation and Animation", Computer Science Department, University of Toronto (M.Sc thesis).

31. Farin, G.., "Curves and Surfaces for Computer Aided Geometric Design", Academic Press, New York, 1988.

32. Faux, I.D and Pratt, M.J., "Computational Geometry for Design and Manufacture", Ellis Horwood Ltd, Chichester, 1983

33. Feiner, S., Salesin, D and Banchoff T., "Dial: A Diagrammatic Animation Language", IEEE Computer Graphics and Applications, 2(9), 1982, pp 43-54.

34. Foley, J and Van Dam, A., "Fundamentals of Interactive Computer Graphics", Addison-Wesley, 1992.

35. Freeman, W.T and Adelson, E.H., "Motion without movement", Proc. ACM SIGGRAPH'91 pp 27-30.

36. Girad, M., "Interactive Design of 3D Computer Animation of Legged Animal Motion", IEEE Computer Graphics and Applications, 7(6), June 1987, pp 39-51.

37. Girad, M and Macienjeuski, A.A., "Computational Modelling for the Computer Animation of Legged Figures", Proc SIGGRAPH 85, pp 263-270.

38. Fox, W., "Computer Animation With Color Registers", BYTE, 1982, pp 194-214.

39. Goldstein, R.A., "A System for Computer Animation of 3D Objects", Proc. 10th UAIDE Annual Meeting, 1971, pp 3-128 to 3-139.

40. Hackathorn, R., "ANIMA II:A 3D Color Animation System", Proc. SIGGRAPH'77, Computer Graphics, 11(2), 1977, pp 54-64.

41. Hackathorn, R., Parent R., Marshall B, Howard M., "An Interactive Microcomputer based 3D Animation System", Proc. Canadian Man-Computer Communications Society Conference'81, 1981, pp 181-191.

42. Halas, J(ed)., "Computer Animation", Hastings House, New York, 1974

43. Herbison-Evans, D., "Real-time Animation of Human Figure Drawings With Hidden-lines Omitted", IEEE Computer Graphics and Applications, 2(9), 1982, pp 27-33.

44. Herbison-Evans, D., "Manipulating Ellipsiods in Animation", Computer Graphics World 7, 1983b, pp 78-82.

45. Hill Jr, F.S., "Computer Graphics", Macmillan, New York, 1990.

46. Honey, F.J., "Artist Oriented Computer Animation", Journal of Society of Motion Picture and Television Engineers, 80(3), 1971, pp 154.

47. HP company, "Starbase Graphics Techniques HP-UX Concepts and Tutorials", vol 1, 2 and 3, Hewlett Pacard Company, 1989.

48. Huggins W.H., Entwisle D.R., "Computer Animation for The Academic Community", SJCC, AFIPS Conference Proceedings, vol 34, 1969, pp 623.

49. Hunter G.M., "Computer Animation Survey", Comput. and Graphics, Pergamon Press, 2, 1977, pp 225-229.

50. Kallis SA., "Computer Animation Techniques", Journal of the SMPTE, 80(3), 1971, pp 145-148.

51. Kochanek, D and Bartels, R., "Interpolating Splines with Local Tension, Continuity and Bias ", Proc. SIGGRAPH 84, Computer Graphics, vol 18, no 3, pp 33-41.

52. Korein J and Badler N., "Temporal Anti-aliasing in Computer Generated Animation", Proc. SIGGRAPH'83, Computer Graphics, 17(3), 1983, pp 377-388.

53. Lansdown, J., " Computer Aided Animation: A Concise Review", Proc. Computer Graphics'82, Online Conf, 1982, pp 267-275.

54. Lassiter, J., "Principles of Traditional Animation Applied to 3D Computer Animation", Proc. SIGGRAPH 87, pp 35-44.

55. Lansdown, J., "The Economics of Computer-aided Animation", Proc. Computer Graphics'83, Online Conf, 1983, pp 267-275.

56. Laybourne, K., "The Animation Book", Crown, 1988.

57. Levoy M., "A Color Animation System Based on the Multiplane Technique", Proc. SIGGRAPH'77, Computer Graphics, 11(2), 1977, pp 65-71.

58. Magnenat-Thalmann, N and Thalmann, D., "Computer Animation: Theory and Practice", Springer-Verlag, Tokyo, 1985.

59. Magnenat-Thalmann, N and Thalmann, D., "Computer Animation '90", Springer-Verlag, Tokyo, 1985.

60. Magnenat-Thalmann N, Thalmann D., "3D Computer Animation Films Using a Programming Language and Interactive Systems", Proc. Computer Graphics'83, Online Conf, 1983, pp 247-257.

61. Magnenat-Thalmann N, Thalmann D., "CINEMIRA: A 3D Computer Animation Language Based on Actor and Camera Data Types", Technical Report, University of Montreal, 1984.

62. Magnenat-Thalmann N, Thalmann D., " 3D Shaded Director-oriented Computer Animation", Proc. Graphics Interface'84, Ottawa, 1984.

63. Magnenat-Thalmann, N and Thalmann, D (eds)., " New trends in Computer Graphics", Proc. CG international '88, Springer-Verlag, New York, 1988.

64. Mäntyalä, M., "An Introduction to Solid Modelling", Computer Science Press, Rockville, MD, 1988.

65. Max, N and Blunden J., "Optical Printing in Computer Animation", Proc. SIGGRAPH'80, Computer Graphics, 14(3), 1980, pp 171-177.

66. Misra, B, K and Pradhan, B.S.S and Dhande, S.G., " Computational Conjugate Geometry of Flexible Generating Curves", Fifth International Conference on Engg. Computer Graphics and Descriptive Geometry, RMTI, Melbourne, Australia, Aug17-21, 1991.

67. Mortenson, M.E., "Geometric Modelling", John Wiley and Sons, New York, 1964.

68. Mezei, L and Zivian A., "ARTA: An Interactive Animation System", Proc. Information Processing 71, North-Holland, 1971, pp 429-434.

69. Mudur, S.P and Patnaik, S.N (eds)., "Graphics, Design and Visualization", Proc. ICCG 93, Tutorial Conference on Computer Graphics, Feb 24-26, 1993, India.

70. Mudur, S.P and Singh , J.H., "A Notation for Computer Animation", IEEE Trans.on Systems, Man and Cybernetics, 8(4), 1978, pp 308-311.

71. Negroponte, N and Pangaro P., "Experiments with Computer Animation", Computer Graphics, 10(2), 1976, pp 40-44.

72. Parke, F.I., "Animation of Faces", Proc. ACM Annual Conf. Vol 1

73. Parke, F.I., "A Model for Human Faces that allows Speech Synchronized Animation", Computers and Graphics, Pergamon Press, 1(1), 1975, pp 1-4.

74. Parke, F.I., "Parameterized Models for Facial Animation", IEEE Computer Graphics and Applications, 2(9), 1982, pp 61-68.

75. Platt, S and Badler N., "Animating Facial Expressions", Proc. SIGGRAPH'81, Computer Graphics, 15(3), 1981, pp 245-252.

76. Potmesil M and Chakravarty I., "Modeling Motion Blur in Computer-generated Images", Proc. SIGGRAPH'83, Computer Graphics, 17(3), 1983, pp 389-399.

77. Raibert, M.H and Hodgins, J.K., "Animation of Dynamic Legged Locomotion", Proc. SIGGRAPH 91, pp 349-358.

78. Reddy, C.C.S., "Animation by Dynamic Attribute Modelling", M.tech. Thesis, Dept. Comp.Sci, I.I.T., Kanpur, 1989.

79. Reeves , W.T., "In-betweening for Computer Animation Utilizing Moving Point Constraints", Proc. SIGGRAPH'81, ACM, 1981, pp 263-269.

80. Reynolds, C.W., "Computer Animation with Scripts and Actors", Proc. SIGGRAPH'82, Computer Graphics, 16(3), 1982, pp 289-296.

81. Riesenfeld R.F. et al., "Using the Oslo Algorithm as a Basis for CAD/CAM

Geometric Modelling", Proc. NCGA'81, National Computer Graphics Association, 1981, pp 345-356.

82. Rijpkema, H and Girad, M., "Computer Animation of Knowledge based Human Grasping", Proc. SIGGRAPH 91, pp 349-358.

83. Rogers, D.F., "Procedural Elements of Computer Graphics", McGraw-Hill, New York, 1985.

84. Russett, R and Starr C., "Experimental Animation", Van Nostrand Reinhold, New York, 1976.

85. Schachter, B.J., "Computer Image Generation for Flight Simulation", IEEE Computer Graphics and Applications, 1(4), 1981, pp 29-68.

86. Schachter, B.J., "Generation of special effects in Computer Image Generation, John Wiley, New York, 1983, pp 155-172.

87. Shoemake, K., "Animating Rotation with Quaternion Curves", Proc. SIGGRAPH 85, pp 245-254.

88. Shoup, R.G., "Colour Table Animation", Proc. SIGGRAPH'79, Computer Graphics, 13(2), 1979, pp 8-13.

89. Sukavirya, P and Moren, L., " Dynamic Construction of Animated Help from Application Context", Proc. SIGGRAPH 88, pp 190-202.

90. Symbolics Inc., "MACSYMA Reference Manual, version 13", Computer Aided Mathematics Group, Symbolics Inc., Birlington, N.A., 1988.

91. Thalmann, D., Magnenat-Thalmann, N and Bergeron P., "Dream Flight: A Fictional Film Produced by 3D Computer Animation", Proc. Computer Graphics'82, online Conf., 1982, pp 353-368.

92. Wein , M and Burtnyk N., "Computer Animation ", In:Encyclopedia of Computer Science and Technology, vol 5, Marcel Dekker, 1976, pp 397-436.

93. Williams, L., "Overview of 3D Animation", Tutorial notes on Computer Animation SIGGRAPH'83, 1983, pp 212-219.

94. Willmert, K.D., "Graphic Display of Human Motion", Proc. ACM'78 Conf., 1978, pp 715-719.

95. Wijchert, J., and Haumann, D., "Animation Aerodynamics", Proc. SIGGRAPH 91, pp 19-22.

96. Wilhelms, J., " Using Dynamic Analysis for Realistics Animation of Articulated Bodies", Computer Graphics and Applications, 2(11), 1982, pp 53-59.

97. Yap, K.T., "Computational Geometry: An Introduction with Applications to NC and Robotics", Addison-Wesley, 1988.

98. Zajac , E.E., "Computer Animation: A New Scientific and Educational Tool", JSMPTE'74, 1965, pp 1006-1008.

99. Zeltzer, D., "Motor Control Techniques for Figure Animation", IEEE Computer Graphics and Applications, 2(9), 1982, pp 53-59.

100. Zeltzer, D., "Representation of Complex Animated Figures", Proc. Graphics Interface'82, 1982b, pp 205-211.

101. Zeleznik, R.C et al., "An Object Oriented Framework for the Integration of Interactive Animation Techniques", Proc. SIGGRAPH 91, pp 105-112.

102. Zeltzer, D., "Knowledge-based Animation", proc.SIGGRAPH/SIGART Workshop on Motion, 1983, pp 187-192.

103. Zimmerlin, R., Stanley J, Stone W, "A Sensor Simulation Animation System", Proc. SIGGRAPH'78, Computer Graphics, 12(3), 1978, pp 105-110.